In compliance with the
Canadian Privacy Legislation
some supporting forms
may have been removed from
this dissertation.

While these forms may be included
in the document page count,
their removal does not represent
any loss of content from the dissertation.

University of Alberta

Scheduling Algorithms for High Throughput Packet Data Service in Cellular Radio Systems

by

Robert Charles Elliott ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Science

Department of Electrical and Computer Engineering

Edmonton, Alberta
Fall 2003

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisisitons et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-87756-6
Our file Notre référence
ISBN: 0-612-87756-6

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou aturement reproduits sans son autorisation.

Canada

University of Alberta

Library Release Form

Name of Author: Robert Charles Elliott

Title of Thesis: Scheduling Algorithms for High Throughput Packet Data Service in Cellular Radio Systems

Degree: Master of Science

Year this Degree Granted: 2003

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

July 2, 2003

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Scheduling Algorithms for High Throughput Packet Data Service in Cellular Radio Systems** submitted by **Robert Charles Elliott** in partial fulfillment of the requirements for the degree of **Master of Science**.

June 25, 2003

Abstract

This work examines the performance of a number of scheduling algorithms for the wireless packet data evolution of third generation cellular systems. A brief description of some of these systems is given. A description of the investigated algorithms (some pre-existing, some newly proposed) and the simulation structure used is provided. The algorithms are then analyzed in three different wireless channel models (two pedestrian, one vehicular). For each channel, both outdated channel state information plus margins and perfect prediction are used to determine the supportable bit rate and transmission format for each user in the sector. Various performance measures including the average throughput per sector and per user, effective bit rates of packets, delay experienced per packet and per user, and the number of time slots allocated to each user are considered and compared between the algorithms, channels, and format determination methods. Conclusions are provided and possible future work is discussed.

To my parents, who most likely understand very little of the material in this
thesis, but have nonetheless supported me in all my academic endeavors

Acknowledgement

I would like to thank Dr. Witold A. Krzymień for his guidance and support in his role as my supervisor during the preparation of this thesis. I would also like to thank Ms. Ge Li of the University of Alberta / TR*Labs*, who conducted the AWGN channel simulations to create the packet error rates curves, and Mr. Carlos Rentel (Qualcomm, Inc.) who performed the initial work on this project. Comments and suggestions were also provided by Mr. Robert Novak and Mr. David Mazzarese, both of the University of Alberta / TR*Labs*.

Funding provided by the Natural Sciences and Engineering Research Council (NSERC) of Canada, the Alberta Informatics Circle of Research Excellence (iCORE), the University of Alberta, and TR*Labs* is gratefully acknowledged.

Finally, I would like to thank my parents, who have supported me throughout my academic career.

Table of Contents

Chapter 1: Introduction 1

1.1. Background and Motivation *1*

1.2. Objectives and Overview of Thesis *4*

Chapter 2: High Throughput Downlink Packet Data Access Systems 5

2.1. Introduction *5*

2.2. 1xEV-DO *6*

2.3. 1xEV-DV *12*

2.4. Conclusion *16*

Chapter 3: Description of Scheduling Algorithms 17

3.1. Introduction *17*

3.2. Round Robin (RR) *18*

3.3. Maximum Rate / Maximum DRC (maxD) *18*

3.4. Proportionally Fair (PF) *19*

3.5. Maximum Rate / Proportionally Fair Hybrid (maxD/PF) *21*

3.6. Modified Largest Weighted Delay First (M-LWDF) *21*

3.7. Slot-wise Fair and Packet-wise Fair (slot-fair & packet-fair) *22*

3.8. Minimum Performance (minperf) *23*

3.9. The Exponential Rule (EXP) *24*

3.10. The maxD/PF/EXP Hybrid *25*

Chapter 4: System Model and Simulation Structure Used for Performance Evaluation 26

4.1. Introduction 26

4.2. Link-Level Simulations 26

4.3. Radio Environment Simulations 29

4.4. System Level Simulations 47

Chapter 5: Performance Evaluation 52

5.1. Introduction 52

5.2. Process of Deciding to Use Individual, Iterative Margins per Transmission Format 54

5.3. Pedestrian Channels, No Prediction (i.e. Using Current Estimated SIRs + Margins) 71

5.3.1. Round Robin (RR), Maximum Rate/DRC (maxD), and Proportionally Fair (PF A & B) 71

5.3.2. Maximum Rate / Proportionally Fair Hybrid (maxD/PF) 91

5.3.3. Modified Largest Weighted Delay First (M-LWDF 1 & 2) 110

5.3.4. Slot-wise Fair ("slot-fair") and Packet-wise Fair ("packet-fair") 125

5.3.5. Minimum Performance (minperf) 142

5.3.6. Minimum Performance ("PF B" method) 160

5.3.7. Exponential Rule (EXP1A, EXP2A, EXP1B, & EXP2B) 177

5.3.8. Maximum Rate / Proportionally Fair / Exponential Rule Hybrid (maxD/PF/EXP) 194

5.3.9. Conclusions 207

5.4. Pedestrian Channels, Perfect Multislot Prediction 208

5.4.1. Effects of Perfect Multislot Prediction 208

5.4.2. "PF A" and "PF B" Average Rate Calculation Comparisons 220

5.5. ITU Vehicular B Channel, Outdated SIR$_p$s Plus Margins 225

5.5.1. Results from the ITU vehicular B channel 225

5.5.2. "PF A" vs. "PF B" Performance Comparison 235

5.6. ITU Vehicular B Channel, Perfect One-slot Prediction and Margins 240

5.6.1. Effects of Perfect One-slot Prediction 240

5.6.2. "PF A" vs. "PF B" Average Rate Calculation Performance 248

5.7. ITU Vehicular B Channel, Perfect Multislot Prediction 250

5.7.1. Effects of Perfect Multislot Prediction 250

5.7.2. "PF A" vs. "PF B" Average Rate Calculation Performance 257

Chapter 6: Summary of Work, Conclusions, and Further Work 259

6.1. Summary and Conclusions 259

6.2. Contributions and Future Work 265

Bibliography 267

Appendix A: AWGN Packet Error Rate Curves for 1xEV-DO 271

Appendix B: Pedestrian Channel Results using Outdated Channel Information and Margins 274

Appendix C: Pedestrian Channel Results using Perfect Multislot Prediction 296

Appendix D: Pedestrian Channel Results Comparing the "PF A" and "PF B" Average Rate Calculation Methods using Perfect Multislot Prediction 320

Appendix E: ITU Vehicular B Channel Results using Outdated SIR$_p$ Values and Margins 344

Appendix F: ITU Vehicular B Channel Results Comparing the "PF A" and "PF B" Average Rate Calculation Methods using Outdated SIR$_p$ Values and Margins 356

Appendix G: ITU Vehicular B Channel Results using Perfect One-slot Prediction and Margins . 369

Appendix H: ITU Vehicular B Channel Results Comparing the "PF A" and "PF B" Average Rate Calculation Methods using Perfect One-slot Prediction and Margins 381

Appendix I: ITU Vehicular B Channel Results using Perfect Multislot Prediction 391

Appendix J: ITU Vehicular B Channel Results Comparing the "PF A" and "PF B" Average Rate Calculation Methods using Perfect Multislot Prediction 403

Appendix K: Derivation of Probability Distribution of Users at a Given Distance from a Sector Vertex 415

List of Tables

Table I: 1xEV-DO Traffic Channel Transmission Formats 7

Table II: Possible Encoder Packet Sizes, Subpacket Data Rates, and Range of Subpacket Sizes for the 1xEV-DV System 13

Table III: Possible Received Packet Formats for 1xEV-DO 27

Table IV: Values of Parameters Used for Cell Radius Computation 30

Table V: Link Budget Values for the Forward Channel 33

Table VI: Delay and Power Profiles of the ITU pedestrian A, ITU pedestrian B, and ITU vehicular B Channels 42

Table VII: The *SIR*$_I$ Resulting from System Imperfections 48

Table VIII: Required E_S/N_0 to Obtain 1 % PER in an AWGN Channel 50

Table IX: Iterative Margins per DRC Based on 1% Level from SIR CDFs 62

Table X: Iterative Margins per DRC Based on 5% Level from SIR CDFs 63

Table XI: Average Packet Error Rates per Format Using RR Algorithm in ITU Pedestrian A Channel Using Variable Margins per Format Plus Bias to Obtain 1% PER per User vs. Using Iterative Margins per Format 65

Table XII: Statistics for Average Throughput per User per Drop Distributions for the Round Robin, Maximum Rate, and Proportionally Fair Scheduling Algorithms in the ITU Pedestrian A and B Channels with 16 Users per Sector 74

Table XIII: Percentiles and Standard Deviations of Packet Delays for the Round Robin, Maximum Rate, and Proportionally Fair Scheduling Algorithms in the ITU Pedestrian A and B channels with 16 Users per Sector 85

Table XIV: Percentiles and Standard Deviation of Average Packet Delay per User for the Round Robin, Maximum Rate, and Proportionally Fair Scheduling Algorithms in the ITU Pedestrian A and B Channels with 16 Users per Sector 87

Table XV: Percentiles and Standard Deviation of Slots Allocated per User per Drop for the Round Robin, Maximum Rate, and Proportionally Fair Scheduling Algorithms in the ITU Pedestrian A and B Channels with 16 Users per Sector 89

Table XVI: Statistics for Average Throughput per User per Drop Distributions for the Maximum Rate / Proportionally Fair Hybrid Scheduling Algorithm in the ITU Pedestrian A and B Channels with 16 Users per Sector 98

Table XVII: Percentiles and Standard Deviations of Packet Delays for the maxD/PF Scheduling Algorithm in the ITU Pedestrian A and B channels with 16 Users per Sector 106

Table XVIII: Percentiles and Standard Deviation of Average Packet Delay per User for the maxD/PF Scheduling Algorithm in the ITU Pedestrian A and B Channels with 16 Users per Sector 108

Table XIX: Percentiles and Standard Deviation of Slots Allocated per User per Drop for the maxD/PF Scheduling Algorithm in the ITU Pedestrian A and B Channels with 16 Users per Sector 110

Table XX: Statistics for Average Throughput per User per Drop Distributions for the M-LWDF 1 & 2 Scheduling Algorithms in the ITU Pedestrian A and B Channels with 16 Users per Sector 113

Table XXI: Percentiles and Standard Deviations of Packet Delays for the M-LWDF Scheduling Algorithms in the ITU Pedestrian A and B channels with 16 Users per Sector 120

Table XXII: Percentiles and Standard Deviation of Average Packet Delay per User for the M-LWDF Scheduling Algorithms in the ITU Pedestrian A and B Channels with 16 Users per Sector 122

Table XXIII: Percentiles and Standard Deviation of Slots Allocated per User per Drop for the M-LWDF Scheduling Algorithms in the ITU Pedestrian A and B Channels with 16 Users per Sector 124

Table XXIV: Statistics for Average Throughput per User per Drop Distributions for the Slot-wise Fair and Packet-wise Fair Scheduling Algorithms in the ITU Pedestrian A and B Channels with 16 Users per Sector 130

Table XXV: Percentiles and Standard Deviations of Packet Delays for the Slot-wise Fair and Packet-wise Fair Scheduling Algorithms in the ITU Pedestrian A and B channels with 16 Users per Sector 138

Table XXVI: Percentiles and Standard Deviation of Average Packet Delay per User for the Slot-wise Fair and Packet-wise Fair Scheduling Algorithms in the ITU Pedestrian A and B Channels with 16 Users per Sector 140

Table XXVII: Percentiles and Standard Deviation of Slots Allocated per User per Drop for the Slot-wise Fair and Packet-wise Fair Scheduling Algorithms in the ITU Pedestrian A and B Channels with 16 Users per Sector 142

Table XXVIII: Statistics for Average Throughput per User per Drop Distributions and Probability of Not Achieving Minimum Specified Throughput for the Minimum Performance Scheduling Algorithm in the ITU Pedestrian A and B Channels with 16 Users per Sector ... 146

Table XXIX: Percentiles and Standard Deviations of Packet Delays for the Minimum Performance Scheduling Algorithm in the ITU Pedestrian A and B channels with 16 Users per Sector 156

Table XXX: Percentiles and Standard Deviation of Average Packet Delay per User for the Minimum Performance Scheduling Algorithm in the ITU Pedestrian A and B Channels with 16 Users per Sector 158

Table XXXI: Percentiles and Standard Deviation of Slots Allocated per User per Drop for the Minimum Performance Scheduling Algorithm in the ITU Pedestrian A and B Channels with 16 Users per Sector 159

Table XXXII: Statistics for Average Throughput per User per Drop Distributions and Probability of Not Achieving Minimum Specified Throughput for the Minimum Performance Scheduling Algorithm in the ITU Pedestrian A Channel with 16 Users per Sector and Using the "PF A" and "PF B" Methods to Calculate the Average Rate 163

Table XXXIII: Statistics for Average Throughput per User per Drop Distributions and Probability of Not Achieving Minimum Specified Throughput for the Minimum Performance Scheduling Algorithm in the ITU Pedestrian B Channel with 16 Users per Sector and Using the "PF A" and "PF B" Methods to Calculate the Average Rate 164

Table XXXIV: Percentiles and Standard Deviations of Packet Delays for the Minimum Performance Scheduling Algorithm in the ITU Pedestrian A Channel with 16 Users per Sector and Using the "PF A" and "PF B" Methods to Calculate the Average Rate 172

Table XXXV: Percentiles and Standard Deviations of Packet Delays for the Minimum Performance Scheduling Algorithm in the ITU Pedestrian B Channel with 16 Users per Sector and Using the "PF A" and "PF B" Methods to Calculate the Average Rate 172

Table XXXVI: Percentiles and Standard Deviation of Average Packet Delay per User for the Minimum Performance Scheduling Algorithm in the ITU Pedestrian A Channel with 16 Users per Sector and Using the "PF A" and "PF B" Methods to Calculate the Average Rate 174

Table XXXVII: Percentiles and Standard Deviation of Average Packet Delay per User for the Minimum Performance Scheduling Algorithm in the ITU Pedestrian B Channel with 16 Users per Sector and Using the "PF A" and "PF B" Methods to Calculate the Average Rate 174

Table XXXVIII: Percentiles and Standard Deviation of Slots Allocated per User per Drop for the Minimum Performance Scheduling Algorithm in the ITU Pedestrian A Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate 176

Table XXXIX: Percentiles and Standard Deviation of Slots Allocated per User per Drop for the Minimum Performance Scheduling Algorithm in the ITU Pedestrian B Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate 176

Table XL: Statistics for Average Throughput per User per Drop Distributions for the Exponential Rule Scheduling Algorithm in the ITU Pedestrian A and B Channels with 16 Users per Sector 181

Table XLI: Percentiles and Standard Deviations of Packet Delays for the Exponential Rule Scheduling Algorithm in the ITU Pedestrian A and B channels with 16 Users per Sector 190

Table XLII: Percentiles and Standard Deviation of Average Packet Delay per User for the Exponential Rule Scheduling Algorithm in the ITU Pedestrian A and B Channels with 16 Users per Sector....192

Table XLIII: Percentiles and Standard Deviation of Slots Allocated per User per Drop for the Exponential Rule Scheduling Algorithm in the ITU Pedestrian A and B Channels with 16 Users per Sector....194

Table XLIV: Statistics for Average Throughput per User per Drop Distributions for the maxD/PF/EXP Scheduling Algorithm in the ITU Pedestrian A and B Channels with 16 Users per Sector....196

Table XLV: Percentiles and Standard Deviations of Packet Delays for the maxD/PF/EXP Scheduling Algorithm in the ITU Pedestrian A and B channels with 16 Users per Sector....204

Table XLVI: Percentiles and Standard Deviation of Slots Allocated per User per Drop for the maxD/PF/EXP Scheduling Algorithm in the ITU Pedestrian A and B Channels with 16 Users per Sector....207

Table XLVII: Mean, Median, and Standard Deviation of Average Throughput per User per Drop Distributions for Scheduling Algorithms in the ITU Pedestrian A Channel with 16 Users/Sector Comparing Use of Margins with Perfect Multislot Prediction....211

Table XLVIII: Mean, Median, and Standard Deviation for Scheduling Algorithms in the ITU Pedestrian B Channel with 16 Users/Sector Comparing Use of Margins with Perfect Multislot Prediction....212

Table XLIX: Percentiles and Standard Deviations of Packet Delays for Scheduling Algorithms in the ITU Pedestrian A Channel with 16 Users per Sector Comparing Use of Margins with Perfect Multislot Prediction....214

Table L: Percentiles and Standard Deviations of Packet Delays for Scheduling Algorithms in the ITU Pedestrian B Channel with 16 Users per Sector Comparing Use of Margins with Perfect Multislot Prediction....215

Table LI: Percentiles and Standard Deviations of Average Packet Delay per User for Scheduling Algorithms in the ITU Pedestrian A Channel with 16 Users per Sector Comparing Use of Margins with Perfect Multislot Prediction....216

Table LII: Percentiles and Standard Deviations of Average Packet Delay per User for Scheduling Algorithms in the ITU Pedestrian B Channel with 16 Users per Sector Comparing Use of Margins with Perfect Multislot Prediction....217

Table LIII: Percentiles and Standard Deviations of Allocated Slots per User per Drop for Scheduling Algorithms in the ITU Pedestrian A Channel with 16 Users per Sector Comparing Use of Margins with Perfect Multislot Prediction....218

Table LIV: Percentiles and Standard Deviations of Allocated Slots per User per Drop for Scheduling Algorithms in the ITU Pedestrian B Channel with 16 Users per Sector Comparing Use of Margins with Perfect Multislot Prediction....218

Table LV: Statistics for Average Throughput per User per Drop Distributions and Probability of Not Achieving Minimum Specified Throughput for the Minimum Performance Scheduling Algorithm in the ITU Pedestrian A Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate and Perfect Multislot Prediction 221

Table LVI: Statistics for Average Throughput per User per Drop Distributions and Probability of Not Achieving Minimum Specified Throughput for the Minimum Performance Scheduling Algorithm in the ITU Pedestrian B Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate and Perfect Multislot Prediction 221

Table LVII: Percentiles and Standard Deviations of Packet Delays for the Minimum Performance Scheduling Algorithm in the ITU Pedestrian A Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate and Perfect Multislot Prediction 222

Table LVIII: Percentiles and Standard Deviations of Packet Delays for the Minimum Performance Scheduling Algorithm in the ITU Pedestrian B Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate and Perfect Multislot Prediction 222

Table LIX: Percentiles and Standard Deviation of Average Packet Delay per User for the Minimum Performance Scheduling Algorithm in the ITU Pedestrian A Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate and Perfect Multislot Prediction..............223

Table LX: Percentiles and Standard Deviation of Average Packet Delay per User for the Minimum Performance Scheduling Algorithm in the ITU Pedestrian B Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate and Perfect Multislot Prediction..............223

Table LXI: Percentiles and Standard Deviation of Slots Allocated per User per Drop for Scheduling Algorithms in the ITU Pedestrian A Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate and Perfect Multislot Prediction 224

Table LXII: Percentiles and Standard Deviation of Slots Allocated per User per Drop for Scheduling Algorithms in the ITU Pedestrian B Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate and Perfect Multislot Prediction 224

Table LXIII: Mean, Median, and Standard Deviation of Average Throughput per User per Drop Distributions for Scheduling Algorithms in the ITU Vehicular B Channel with 16 Users per Sector..............228

Table LXIV: Percentiles and Standard Deviations of Packet Delays for Scheduling Algorithms in the ITU Vehicular B Channel with 16 Users per Sector..............231

Table LXV: Percentiles and Standard Deviations of Average Packet Delay per User for Scheduling Algorithms in the ITU Vehicular B Channel with 16 Users per Sector for Users who Receive at Least One Packet per Drop 233

Table LXVI: Percentiles and Standard Deviations of Allocated Slots per User per Drop for Scheduling Algorithms in the ITU Vehicular B Channel with 16 Users per Sector............234

Table LXVII: Mean, Median, and Standard Deviation of Average Throughput per User per Drop Distributions for Scheduling Algorithms in the ITU Vehicular B Channel with 16 Users per Sector Comparing the "PF A" and "PF B" Methods of Calculating the Average Rate.......237

Table LXVIII: Percentiles and Standard Deviations of Packet Delays for the Minimum Performance Scheduling Algorithm in the ITU Vehicular B Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate...................238

Table LXIX: Percentiles and Standard Deviations of Average Packet Delay per User per Drop for Scheduling Algorithms in the ITU Vehicular B Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate 239

Table LXX: Percentiles and Standard Deviations of Slots Allocated per User per Drop for Scheduling Algorithms in the ITU Vehicular B Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate 240

Table LXXI: Mean, Median, and Standard Deviation of Average Throughput per User per Drop Distributions for Scheduling Algorithms in the ITU Vehicular B Channel with 16 Users per Sector and Perfect One-slot Prediction...243

Table LXXII: Percentiles and Standard Deviations of Packet Delays for Scheduling Algorithms in the ITU Vehicular B Channel with 16 Users per Sector and Perfect One-slot Prediction...244

Table LXXIII: Percentiles and Standard Deviations of Average Packet Delay per User for Scheduling Algorithms in the ITU Vehicular B Channel with 16 Users per Sector and Perfect One-slot Prediction..245

Table LXXIV: Percentiles and Standard Deviations of Allocated Slots per User per Drop for Scheduling Algorithms in the ITU Vehicular B Channel with 16 Users per Sector and Perfect One-slot Prediction..246

Table LXXV: Statistics for Average Throughput per User per Drop Distributions and Probability of Not Achieving Minimum Specified Throughput for the Minimum Performance Scheduling Algorithm in the ITU Vehicular B Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate and Perfect One-slot Prediction 248

Table LXXVI: Percentiles and Standard Deviations of Packet Delays for the Minimum Performance Scheduling Algorithm in the ITU Vehicular B Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate and Perfect One-slot Prediction...249

Table LXXVII: Percentiles and Standard Deviation of Average Packet Delay per User for the Minimum Performance Scheduling Algorithm in the ITU Vehicular B Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate and Perfect One-slot Prediction..249

Table LXXVIII: Percentiles and Standard Deviation of Slots Allocated per User per Drop for the Minimum Performance Scheduling Algorithm in the ITU Vehicular B Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate and Perfect One-Slot Prediction 249

Table LXXIX: Mean, Median, and Standard Deviation of Average Throughput per User per Drop Distributions for Scheduling Algorithms in the ITU Vehicular B Channel with 16 Users per Sector and Perfect Multislot Prediction 252

Table LXXX: Percentiles and Standard Deviations of Packet Delays for Scheduling Algorithms in the ITU Vehicular B Channel with 16 Users per Sector and Perfect Multislot Prediction 253

Table LXXXI: Percentiles and Standard Deviations of Average Packet Delay per User for Scheduling Algorithms in the ITU Vehicular B Channel with 16 Users per Sector and Perfect Multislot Prediction 255

Table LXXXII: Percentiles and Standard Deviations of Allocated Slots per User per Drop for Scheduling Algorithms in the ITU Vehicular B Channel with 16 Users per Sector and Perfect Multislot Prediction 256

Table LXXXIII: Statistics for Average Throughput per User per Drop Distributions and Probability of Not Achieving Minimum Specified Throughput for the Minimum Performance Scheduling Algorithm in the ITU Vehicular B Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate and Perfect Multislot Prediction 257

Table LXXXIV: Percentiles and Standard Deviations of Packet Delays for the Minimum Performance Scheduling Algorithm in the ITU Vehicular B Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate and Perfect Multislot Prediction 257

Table LXXXV: Percentiles and Standard Deviation of Average Packet Delay per User for the Minimum Performance Scheduling Algorithm in the ITU Vehicular B Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate and Perfect Multislot Prediction 258

Table LXXXVI: Percentiles and Standard Deviation of Slots Allocated per User per Drop for the Minimum Performance Scheduling Algorithm in the ITU Vehicular B Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate and Perfect Multislot Prediction 258

Table LXXXVII: *P* Percent of the Time, At Least *Q* Percent of the Users Out of *N* Users in Total Will Be Located Outside a Radius of *r* Away From the Vertex of the Sector 421

List of Figures

Figure 2.1: 1xEV-DO forward link slot structure 7

Figure 2.2: Signal constellations for (a) QPSK, (b) 8-PSK, and (c) 16-QAM modulations 8

Figure 2.3: Preamble, pilot, MAC, and data multiplexing for the multiple-slot cases with data rates of 153.6, 307.2, 614.4, 921.6, and 1228.8 kb/s 9

Figure 2.4: Preamble, pilot, MAC, and data multiplexing with data rates of 38.4 and 76.8 kb/s 9

Figure 2.5: Preamble, pilot, MAC, and data multiplexing for the 1-slot case with a data rate of 614.4 kb/s 10

Figure 2.6: Preamble, pilot, MAC, and data multiplexing for the 1-slot cases with data rates of 1228.8, 1843.2, and 2457.6 kb/s 10

Figure 2.7: Multiple-slot packet with normal termination 11

Figure 2.8: Multiple-slot packet with early termination 11

Figure 4.1: Block diagram of simulation process 26

Figure 4.2: Cell layout used for simulations 29

Figure 4.3: Directivity loss pattern versus angle from antenna bore sight for EMS Wireless RR65-18-00DP antenna 31

Figure 4.4: (a) Received power relative to the bore sight at 0° in dB versus the angle from the bore sight of the antenna if the mobile receiver is positioned at the border of a sector; (b) The perimeter of a sector, where the measurements for (a) would be taken 32

Figure 4.5: Comparison of simulated and postulated exponential auto-correlation of 760,000 lognormal shadow fading values 36

Figure 4.6: (a) Normalized power spectral density of Rayleigh fading process; (b) Sampled version of normalized power spectral density. f_s = sampling frequency; f_m = Doppler frequency 38

Figure 4.7: Block diagram of the modified Smith method 38

Figure 4.8: Comparison of simulated Rayleigh envelope and power values with theoretical probability density functions 40

Figure 4.9: Simulated and ideal auto-correlation of Rayleigh fading process at a mobile speed of 3 km/h and carrier frequency of 1960 MHz 41

Figure 4.10: Simulated and ideal auto-correlation of Rayleigh fading process at a mobile speed of 60 km/h and carrier frequency of 1960 MHz 41

Figure 4.11: CDFs of SIR values obtained from radio environment simulations 47

Figure 4.12: CDFs of modified SIR values after accounting for system imperfections........ 48

Figure 5.1: Average PER per user vs. number of users per sector for ITU pedestrian A channel using 1 dB margin for format determination 54

Figure 5.2: Average PER per user vs. number of users per sector for ITU pedestrian A channel using 1.1 dB to 3.0 dB margin for format determination........ 55

Figure 5.3: Average PER per user vs. number of users per sector for ITU pedestrian A channel using 4.0 dB margin for format determination 55

Figure 5.4: CDFs for all formats in ITU pedestrian A channel of differences between averaged SIR level over all possible transmission slots and current SIR level, conditioned on current SIR level being above that required for 1% PER........ 57

Figure 5.5: CDFs of $F\{\mathrm{AVG}[\mathrm{SIR}(slot+3),\mathrm{SIR}(slot+7),\ldots,\mathrm{SIR}(slot+63)]-\mathrm{SIR}(slot) \mid \mathrm{SIR}(slot) > (-13.5+m)\ \mathrm{dB}\}$ with iterations of m based on 1% CDF level for DRC1 in ITU pedestrian A channel 58

Figure 5.6: CDFs of $F\{\mathrm{AVG}[\mathrm{SIR}(slot+3),\mathrm{SIR}(slot+7),\ldots,\mathrm{SIR}(slot+31)]-\mathrm{SIR}(slot) \mid \mathrm{SIR}(slot) > (-10.5+m)\ \mathrm{dB}\}$ with iterations of m based on 1% CDF level for DRC2 in ITU pedestrian A channel 58

Figure 5.7: CDFs of $F\{\mathrm{AVG}[\mathrm{SIR}(slot+3),\mathrm{SIR}(slot+7), \mathrm{SIR}(slot+11),\mathrm{SIR}(slot+15)]-\mathrm{SIR}(slot) \mid \mathrm{SIR}(slot) > (-7.4+m)\ \mathrm{dB}\}$ with iterations of m based on 1% CDF level for DRC3 in ITU pedestrian A channel 59

Figure 5.8: CDFs of $F\{\mathrm{AVG}[\mathrm{SIR}(slot+3),\mathrm{SIR}(slot+7)]-\mathrm{SIR}(slot) \mid \mathrm{SIR}(slot) > (-4.3+m)\ \mathrm{dB}\}$ with iterations of m based on 1% CDF level for DRC4 in ITU pedestrian A channel 59

Figure 5.9: CDFs of $F\{\mathrm{SIR}(slot+3)-\mathrm{SIR}(slot) \mid \mathrm{SIR}(slot) > (-1.0+m)\ \mathrm{dB}\}$ with iterations of m based on 1% CDF level for DRC5 in ITU pedestrian A channel 60

Figure 5.10: CDFs of $F\{\mathrm{SIR}(slot+3)-\mathrm{SIR}(slot) \mid \mathrm{SIR}(slot) > (3.7+m)\ \mathrm{dB}\}$ with iterations of m based on 1% CDF level for DRC8 in ITU pedestrian A channel 60

Figure 5.11: CDFs of $F\{\mathrm{AVG}[\mathrm{SIR}(slot+3),\mathrm{SIR}(slot+7)]-\mathrm{SIR}(slot) \mid \mathrm{SIR}(slot) > (1.5+m)\ \mathrm{dB}\}$ with iterations of m based on 1% CDF level for DRC9 in ITU pedestrian A channel 61

Figure 5.12: CDFs of $F\{\mathrm{SIR}(slot+3)-\mathrm{SIR}(slot) \mid \mathrm{SIR}(slot) > (7.1+m)\ \mathrm{dB}\}$ with iterations of m based on 1% CDF level for DRC10 in ITU pedestrian A channel 61

Figure 5.13: CDFs of $F\{\mathrm{SIR}(slot+3)-\mathrm{SIR}(slot) \mid \mathrm{SIR}(slot) > (9.2+m)\ \mathrm{dB}\}$ with iterations of m based on 1% CDF level for DRC12 in ITU pedestrian A channel 62

Figure 5.14: Average PER per user vs. number of users per sector for ITU pedestrian A channel using margins obtained from 1% level on SIR difference CDF curves 63

Figure 5.15: Average PER per user vs. number of users per sector for ITU pedestrian A channel using margins obtained from 5% level on SIR difference CDF curves 64

Figure 5.16: Approximate probability of a DRC1 packet error in the ITU pedestrian A channel, assuming that a DRC1 packet is scheduled to a user with an estimated signal-to-noise ratio of SIR(*slot*) 69

Figure 5.17: PDF of ITU pedestrian A SIR values, conditioned on the values being less than –8.5 dB 70

Figure 5.18: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian A channel for the round robin, maximum rate, and proportionally fair scheduling algorithms 71

Figure 5.19: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian B channel for the round robin, maximum rate, and proportionally fair scheduling algorithms 72

Figure 5.20: Square sectors with groups of 1, 4, 9, and 16 users distributed uniformly over the area of the sector 73

Figure 5.21: Distributions of average throughput per user per drop for the round robin, maximum rate, and proportionally fair algorithms in the ITU pedestrian A channel with 16 users per sector 74

Figure 5.22: Distributions of average throughput per user per drop for the round robin, maximum rate, and proportionally fair algorithms in the ITU pedestrian B channel with 16 users per sector 74

Figure 5.23: Distributions of effective bit rates for packets transmitted under the round robin, maximum rate, and proportionally fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector 76

Figure 5.24: Distributions of effective bit rates for packets transmitted under the round robin, maximum rate, and proportionally fair scheduling algorithms in the ITU pedestrian B channel with 16 users per sector 77

Figure 5.25: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a) round robin, (b) maximum rate, and (c),(d) proportionally fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector 77

Figure 5.26: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a) round robin, (b) maximum rate, and (c),(d) proportionally fair scheduling algorithms in the ITU pedestrian B channel with 16 users per sector 78

Figure 5.27: Average delay per packet per drop versus number of users per sector for the round robin, maximum rate, and proportionally fair scheduling algorithms in the ITU pedestrian A channel 79

Figure 5.28: Average delay per packet per drop versus number of users per sector for the round robin, maximum rate, and proportionally fair scheduling algorithms in the ITU pedestrian B channel 80

Figure 5.29: Average packet delay per user per drop versus number of users per sector for the round robin, maximum rate, and proportionally fair scheduling algorithms in the ITU pedestrian A channel on a (a) linear scale and (b) logarithmic scale....81

Figure 5.30: Average packet delay per user per drop versus number of users per sector for the round robin, maximum rate, and proportionally fair scheduling algorithms in the ITU pedestrian B channel on a (a) linear scale and (b) logarithmic scale....81

Figure 5.31: Cumulative distribution functions of delays experienced by packets under the round robin, maximum rate, and proportionally fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector....83

Figure 5.32: Cumulative distribution functions of delays experienced by packets under the round robin, maximum rate, and proportionally fair scheduling algorithms in the ITU pedestrian B channel with 16 users per sector....83

Figure 5.33: Probability of delays experienced by packets being larger than a given value under the round robin, maximum rate, and proportionally fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector....84

Figure 5.34: Probability of delays experienced by packets being larger than a given value under the round robin, maximum rate, and proportionally fair scheduling algorithms in the ITU pedestrian B channel with 16 users per sector....84

Figure 5.35: Distributions of average packet delay per user for round robin, maximum rate, and proportionally fair scheduling algorithms in ITU pedestrian A channel with 16 users per sector 86

Figure 5.36: Distributions of average packet delay per user for round robin, maximum rate, and proportionally fair scheduling algorithms in ITU pedestrian B channel with 16 users per sector 87

Figure 5.37: Distributions of allocated slots per user per drop for the round robin, maximum rate, and proportionally fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector 88

Figure 5.38: Distributions of allocated slots per user per drop for the round robin, maximum rate, and proportionally fair scheduling algorithms in the ITU pedestrian B channel with 16 users per sector 89

Figure 5.39: Average sector throughput per drop from maxD/PF scheduling algorithm in ITU pedestrian A channel with 16 users per sector and R_{denom}=1248 91

Figure 5.40: Average sector throughput per drop from maxD/PF scheduling algorithm in ITU pedestrian A channel with 16 users per sector and R_{denom}=1248000......92

Figure 5.41: Typical average rate values (units of bits per second) in the proportionally fair scheduling algorithm metric for a drop of 16 users per sector in the ITU pedestrian A channel......93

Figure 5.42: Mean of the average rate values in the proportionally fair scheduling algorithm across 160 users in the ITU pedestrian A channel......93

Figure 5.43: Average sector throughput per drop from maxD/PF scheduling algorithm in ITU pedestrian A channel with 16 users per sector and R_{denom}=105000......94

Figure 5.44: Average sector throughput per drop versus number of users per sector for maxD/PF scheduling algorithm with R_{denom} = 105000 bits per second in ITU pedestrian A channel....94

Figure 5.45: Average sector throughput per drop versus number of users per sector in ITU pedestrian A channel for maxD/PF scheduling algorithm with R_{denom} replaced by $\overline{R}_{av}(t)$...96

Figure 5.46: Average sector throughput per drop versus number of users per sector in ITU pedestrian B channel for maxD/PF scheduling algorithm with $\overline{R}_{av}(t)$ denominator......96

Figure 5.47: Distributions of average throughput per user per drop for the maxD/PF scheduling algorithm in the ITU pedestrian A channel with 16 users per sector......97

Figure 5.48: Distributions of average throughput per user per drop for the maxD/PF scheduling algorithm in the ITU pedestrian B channel with 16 users per sector......98

Figure 5.49: Distributions of effective bit rates for packets transmitted under the maxD/PF hybrid scheduling algorithms in the ITU pedestrian A channel with 16 users per sector......99

Figure 5.50: Distributions of effective bit rates for packets transmitted under the maxD/PF hybrid scheduling algorithms in the ITU pedestrian B channel with 16 users per sector......99

Figure 5.51: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a) maximum rate, (b),(c),(d) maxD/PF hybrid, and (e) proportionally fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector......100

Figure 5.52: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a) maximum rate, (b),(c),(d) maxD/PF hybrid, and (e) proportionally fair scheduling algorithms in the ITU pedestrian B channel with 16 users per sector......100

Figure 5.53: Average delay per packet per drop versus number of users per sector for the maxD/PF scheduling algorithm in the ITU pedestrian A channel......101

Figure 5.54: Average delay per packet per drop versus number of users per sector for the maxD/PF scheduling algorithm in the ITU pedestrian B channel 102

Figure 5.55: Average packet delay per user per drop versus number of users per sector for the maxD/PF scheduling algorithm in the ITU pedestrian A channel 103

Figure 5.56: Average packet delay per user per drop versus number of users per sector for the maxD/PF scheduling algorithm in the ITU pedestrian B channel 103

Figure 5.57: Cumulative distribution functions of delays experienced by packets under the maxD/PF scheduling algorithm in the ITU pedestrian A channel with 16 users per sector.... 104

Figure 5.58: Cumulative distribution functions of delays experienced by packets under the maxD/PF scheduling algorithm in the ITU pedestrian B channel with 16 users per sector.... 105

Figure 5.59: Probability of delays experienced by packets being larger than a given value under the maxD/PF scheduling algorithm in the ITU pedestrian A channel with 16 users per sector 105

Figure 5.60: Probability of delays experienced by packets being larger than a given value under the maxD/PF scheduling algorithm in the ITU pedestrian B channel with 16 users per sector 106

Figure 5.61: Distributions of average packet delay per user for maxD/PF scheduling algorithm in ITU pedestrian A channel with 16 users per sector 107

Figure 5.62: Distributions of average packet delay per user for maxD/PF scheduling algorithm in ITU pedestrian B channel with 16 users per sector 108

Figure 5.63: Distributions of allocated slots per user per drop for the maxD/PF scheduling algorithm in the ITU pedestrian A channel with 16 users per sector.................................... 109

Figure 5.64: Distributions of allocated slots per user per drop for the maxD/PF scheduling algorithm in the ITU pedestrian B channel with 16 users per sector.................................... 109

Figure 5.65: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian A channel for the M-LWDF 1 & 2 scheduling algorithms.................................. 111

Figure 5.66: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian B channel for the M-LWDF 1 & 2 scheduling algorithms.................................. 111

Figure 5.67: Distributions of average throughput per user per drop for the M-LWDF scheduling algorithm in the ITU pedestrian A channel with 16 users per sector.................................... 112

Figure 5.68: Distributions of average throughput per user per drop for the M-LWDF scheduling algorithm in the ITU pedestrian B channel with 16 users per sector.................................... 112

Figure 5.69: Distributions of effective bit rates for packets transmitted under the M-LWDF scheduling algorithms in the ITU pedestrian A channel with 16 users per sector 113

Figure 5.70: Distributions of effective bit rates for packets transmitted under the M-LWDF scheduling algorithms in the ITU pedestrian B channel with 16 users per sector 114

Figure 5.71: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a) M-LWDF 1, (b) M-LWDF 2, and (c) proportionally fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector 114

Figure 5.72: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a) M-LWDF 1, (b) M-LWDF 2, and (c) proportionally fair scheduling algorithms in the ITU pedestrian B channel with 16 users per sector 115

Figure 5.73: Average delay per packet per drop versus number of users per sector for the M-LWDF scheduling algorithms in the ITU pedestrian A channel 116

Figure 5.74: Average delay per packet per drop versus number of users per sector for the M-LWDF scheduling algorithms in the ITU pedestrian B channel 116

Figure 5.75: Average packet delay per user per drop versus number of users per sector for the M-LWDF scheduling algorithms in the ITU pedestrian A channel 117

Figure 5.76: Average packet delay per user per drop versus number of users per sector for the M-LWDF scheduling algorithms in the ITU pedestrian B channel 117

Figure 5.77: Cumulative distribution functions of delays experienced by packets under the M-LWDF scheduling algorithms in the ITU pedestrian A channel with 16 users per sector 118

Figure 5.78: Cumulative distribution functions of delays experienced by packets under the M-LWDF scheduling algorithms in the ITU pedestrian B channel with 16 users per sector 119

Figure 5.79: Probability of delays experienced by packets being larger than a given value under the M-LWDF scheduling algorithms in the ITU pedestrian A channel with 16 users per sector 119

Figure 5.80: Probability of delays experienced by packets being larger than a given value under the M-LWDF scheduling algorithms in the ITU pedestrian B channel with 16 users per sector 120

Figure 5.81: Distributions of average packet delay per user for M-LWDF scheduling algorithms in ITU pedestrian A channel with 16 users per sector 121

Figure 5.82: Distributions of average packet delay per user for M-LWDF scheduling algorithms in ITU pedestrian B channel with 16 users per sector 122

Figure 5.83: Distributions of allocated slots per user per drop for the M-LWDF scheduling algorithms in the ITU pedestrian A channel with 16 users per sector 123

Figure 5.84: Distributions of allocated slots per user per drop for the M-LWDF scheduling algorithms in the ITU pedestrian B channel with 16 users per sector 123

Figure 5.85: Initial average sector throughput per drop vs. number of users per sector results in the ITU pedestrian A channel for the slot-fair and packet-fair scheduling algorithms 125

Figure 5.86: Initial average sector throughput per drop vs. number of users per sector results in the ITU pedestrian B channel for the slot-fair and packet-fair scheduling algorithms 126

Figure 5.87: Average sector throughput per drop vs. number of users per sector results in the ITU pedestrian A channel for the corrected slot-fair and packet-fair scheduling algorithms 128

Figure 5.88: Average sector throughput per drop vs. number of users per sector results in the ITU pedestrian B channel for the corrected slot-fair and packet-fair scheduling algorithms...... 128

Figure 5.89: Distributions of average throughput per user per drop for the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector 129

Figure 5.90: Distributions of average throughput per user per drop for the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian B channel with 16 users per sector 130

Figure 5.91: Distributions of effective bit rates for packets transmitted under the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector 131

Figure 5.92: Distributions of effective bit rates for packets transmitted under the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector 131

Figure 5.93: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a) slot-wise fair, (b) packet-wise fair, (c) proportionally fair, and (d) round robin scheduling algorithms in the ITU pedestrian A channel with 16 users per sector 132

Figure 5.94: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a) slot-wise fair, (b) packet-wise fair, (c) proportionally fair, and (d) round robin scheduling algorithms in the ITU pedestrian B channel with 16 users per sector 132

Figure 5.95: Average delay per packet per drop versus number of users per sector for the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian A channel............................ 133

Figure 5.96: Average delay per packet per drop versus number of users per sector for the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian B channel............................ 134

Figure 5.97: Average packet delay per user per drop versus number of users per sector for the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian A channel 135

Figure 5.98: Average packet delay per user per drop versus number of users per sector for the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian B channel 135

Figure 5.99: Cumulative distribution functions of delays experienced by packets under the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector 136

Figure 5.100: Cumulative distribution functions of delays experienced by packets under the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian B channel with 16 users per sector 136

Figure 5.101: Probability of delays experienced by packets being larger than a given value under the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector 137

Figure 5.102: Probability of delays experienced by packets being larger than a given value under the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian B channel with 16 users per sector 137

Figure 5.103: Distributions of average packet delay per user for slot-fair and packet-fair scheduling algorithms in ITU pedestrian A channel with 16 users per sector 139

Figure 5.104: Distributions of average packet delay per user for slot-fair and packet-fair scheduling algorithms in ITU pedestrian B channel with 16 users per sector 139

Figure 5.105: Distributions of allocated slots per user per drop for the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector 141

Figure 5.106: Distributions of allocated slots per user per drop for the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian B channel with 16 users per sector 141

Figure 5.107: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian A channel for the minperf scheduling algorithm 143

Figure 5.108: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian B channel for the minperf scheduling algorithm 144

Figure 5.109: Distributions of average throughput per user per drop for the minperf scheduling algorithm in the ITU pedestrian A channel with 16 users per sector 145

Figure 5.110: Distributions of average throughput per user per drop for the minperf scheduling algorithm in the ITU pedestrian B channel with 16 users per sector 145

Figure 5.111: Distributions of effective bit rates for packets transmitted under the minperf scheduling algorithm in the ITU pedestrian A channel with 16 users per sector 147

Figure 5.112: Distributions of effective bit rates for packets transmitted under the minperf scheduling algorithm in the ITU pedestrian B channel with 16 users per sector 148

Figure 5.113: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a) maximum rate and minperf algorithm with (b) 5 kb/s, (c) 10 kb/s, (d), 15 kb/s, (e) 20 kb/s, and (f) 25 kb/s minimum throughput specification in the ITU pedestrian A channel with 16 users per sector 149

Figure 5.114: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a) maximum rate and minperf algorithm with (b) 5 kb/s, (c) 10 kb/s, (d), 15 kb/s, (e) 20 kb/s, and (f) 25 kb/s minimum throughput specification in the ITU pedestrian B channel with 16 users per sector 150

Figure 5.115: Average delay per packet per drop versus number of users per sector for the minperf scheduling algorithm in the ITU pedestrian A channel 151

Figure 5.116: Average delay per packet per drop versus number of users per sector for the minperf scheduling algorithm in the ITU pedestrian B channel 151

Figure 5.117: Average packet delay per user per drop versus number of users per sector for the minperf scheduling algorithm in the ITU pedestrian A channel 152

Figure 5.118: Average packet delay per user per drop versus number of users per sector for the minperf scheduling algorithm in the ITU pedestrian B channel 153

Figure 5.119: Cumulative distribution functions of delays experienced by packets under the minperf scheduling algorithm in the ITU pedestrian A channel with 16 users per sector ... 154

Figure 5.120: Cumulative distribution functions of delays experienced by packets under the minperf scheduling algorithm in the ITU pedestrian B channel with 16 users per sector ... 154

Figure 5.121: Probability of delays experienced by packets being larger than a given value under the minperf scheduling algorithm in the ITU pedestrian A channel with 16 users per sector 155

Figure 5.122: Probability of delays experienced by packets being larger than a given value under the minperf scheduling algorithm in the ITU pedestrian B channel with 16 users per sector 155

Figure 5.123: Distributions of average packet delay per user for the minperf scheduling algorithm in ITU pedestrian A channel with 16 users per sector 157

Figure 5.124: Distributions of average packet delay per user for the minperf scheduling algorithm in ITU pedestrian B channel with 16 users per sector 157

Figure 5.125: Distributions of allocated slots per user per drop for the minperf scheduling algorithm in the ITU pedestrian A channel with 16 users per sector 158

Figure 5.126: Distributions of allocated slots per user per drop for the minperf scheduling algorithm in the ITU pedestrian B channel with 16 users per sector 159

Figure 5.127: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian A channel for the minperf scheduling algorithm using the "PF B" method to calculate the average rate....161

Figure 5.128: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian B channel for the minperf scheduling algorithm using the "PF B" method to calculate the average rate....161

Figure 5.129: Distributions of average throughput per user per drop for the minperf scheduling algorithm in the ITU pedestrian A channel with 16 users per sector using the "PF B" method to calculate the average rate....162

Figure 5.130: Distributions of average throughput per user per drop for the minperf scheduling algorithm in the ITU pedestrian B channel with 16 users per sector using the "PF B" method to calculate the average rate....163

Figure 5.131: Distributions of effective bit rates for packets transmitted under the minperf scheduling algorithm in the ITU pedestrian A channel with 16 users per sector using the "PF B" method to calculate the average rate....165

Figure 5.132: Distributions of effective bit rates for packets transmitted under the minperf scheduling algorithm in the ITU pedestrian B channel with 16 users per sector using the "PF B" method to calculate the average rate....165

Figure 5.133: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the minperf algorithm in the ITU pedestrian A channel with 16 users per sector using the (a)-(e) "PF A" and (f)-(j) "PF B" methods to calculate the average rate....166

Figure 5.134: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the minperf algorithm in the ITU pedestrian B channel with 16 users per sector using the (a)-(e) "PF A" and (f)-(j) "PF B" methods to calculate the average rate....167

Figure 5.135: Average delay per packet per drop versus number of users per sector for the minperf scheduling algorithm in the ITU pedestrian A channel using the "PF B" method to calculate the average rate....168

Figure 5.136: Average delay per packet per drop versus number of users per sector for the minperf scheduling algorithm in the ITU pedestrian B channel using the "PF B" method to calculate the average rate....168

Figure 5.137: Average packet delay per user per drop versus number of users per sector for the minperf scheduling algorithm in the ITU pedestrian A channel using the "PF B" method to calculate the average rate....169

Figure 5.138: Average packet delay per user per drop versus number of users per sector for the minperf scheduling algorithm in the ITU pedestrian B channel using the "PF B" method to calculate the average rate....169

Figure 5.139: Cumulative distribution functions of delays experienced by packets under the minperf scheduling algorithm in the ITU pedestrian A channel with 16 users per sector using the "PF B" method to calculate the average rate 170

Figure 5.140: Cumulative distribution functions of delays experienced by packets under the minperf scheduling algorithm in the ITU pedestrian B channel with 16 users per sector using the "PF B" method to calculate the average rate 171

Figure 5.141: Probability of delays experienced by packets being larger than a given value under the minperf scheduling algorithm in the ITU pedestrian A channel with 16 users per sector using the "PF B" method to calculate the average rate 171

Figure 5.142: Probability of delays experienced by packets being larger than a given value under the minperf scheduling algorithm in the ITU pedestrian B channel with 16 users per sector using the "PF B" method to calculate the average rate 172

Figure 5.143: Distributions of average packet delay per user for the minperf scheduling algorithm in ITU pedestrian A channel with 16 users per sector using the "PF B" method to calculate the average rates 173

Figure 5.144: Distributions of average packet delay per user for the minperf scheduling algorithm in ITU pedestrian B channel with 16 users per sector using the "PF B" method to calculate the average rates 174

Figure 5.145: Distributions of allocated slots per user per drop for the minperf scheduling algorithm in the ITU pedestrian A channel with 16 users per sector using the "PF B" method to calculate the average rates 175

Figure 5.146: Distributions of allocated slots per user per drop for the minperf scheduling algorithm in the ITU pedestrian B channel with 16 users per sector using the "PF B" method to calculate the average rates 176

Figure 5.147: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian A channel for the EXP scheduling algorithm 178

Figure 5.148: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian B channel for the EXP scheduling algorithm 179

Figure 5.149: Distributions of average throughput per user per drop for the EXP scheduling algorithm in the ITU pedestrian A channel with 16 users per sector 180

Figure 5.150: Distributions of average throughput per user per drop for the EXP scheduling algorithm in the ITU pedestrian B channel with 16 users per sector 180

Figure 5.151: Distributions of effective bit rates for packets transmitted under the EXP scheduling algorithm in the ITU pedestrian A channel with 16 users per sector 181

Figure 5.152: Distributions of effective bit rates for packets transmitted under the EXP scheduling algorithm in the ITU pedestrian B channel with 16 users per sector 182

Figure 5.153: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a) EXP1A, (b) EXP1B, (c) maxD, (d), EXP2A, (e) EXP2B, and (f) PF scheduling algorithms in the ITU pedestrian A channel with 16 users per sector 183

Figure 5.154: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a) EXP1A, (b) EXP1B, (c) maxD, (d), EXP2A, (e) EXP2B, and (f) PF scheduling algorithms in the ITU pedestrian B channel with 16 users per sector 184

Figure 5.155: Average delay per packet per drop versus number of users per sector for the EXP scheduling algorithm in the ITU pedestrian A channel 185

Figure 5.156: Average delay per packet per drop versus number of users per sector for the EXP scheduling algorithm in the ITU pedestrian B channel 185

Figure 5.157: Distributions of average packet delay per user for the EXP scheduling algorithm in ITU pedestrian A channel with 16 users per sector 186

Figure 5.158: Distributions of average packet delay per user for the EXP scheduling algorithm in ITU pedestrian B channel with 16 users per sector 187

Figure 5.159: Cumulative distribution functions of delays experienced by packets under the EXP scheduling algorithm in the ITU pedestrian A channel with 16 users per sector 188

Figure 5.160: Cumulative distribution functions of delays experienced by packets under the EXP scheduling algorithm in the ITU pedestrian B channel with 16 users per sector 188

Figure 5.161: Probability of delays experienced by packets being larger than a given value under the EXP scheduling algorithm in the ITU pedestrian A channel with 16 users per sector 189

Figure 5.162: Probability of delays experienced by packets being larger than a given value under the EXP scheduling algorithm in the ITU pedestrian B channel with 16 users per sector 189

Figure 5.163: Distributions of average packet delay per user for the EXP scheduling algorithm in ITU pedestrian A channel with 16 users per sector for those users who receive at least one packet during a drop 191

Figure 5.164: Distributions of average packet delay per user for the EXP scheduling algorithm in ITU pedestrian B channel with 16 users per sector for those users who receive at least one packet during a drop 191

Figure 5.165: Distributions of allocated slots per user per drop for the EXP scheduling algorithm in the ITU pedestrian A channel with 16 users per sector 193

Figure 5.166: Distributions of allocated slots per user per drop for the EXP scheduling algorithm in the ITU pedestrian B channel with 16 users per sector 193

Figure 5.167: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian A channel for the maxD/PF/EXP scheduling algorithm 195

Figure 5.168: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian B channel for the maxD/PF/EXP scheduling algorithm....................................195

Figure 5.169: Distributions of average throughput per user per drop for the maxD/PF/EXP scheduling algorithm in the ITU pedestrian A channel with 16 users per sector 196

Figure 5.170: Distributions of average throughput per user per drop for the maxD/PF/EXP scheduling algorithm in the ITU pedestrian B channel with 16 users per sector.................196

Figure 5.171: Distributions of effective bit rates for packets transmitted under the maxD/PF/EXP scheduling algorithm in the ITU pedestrian A channel with 16 users per sector 197

Figure 5.172: Distributions of effective bit rates for packets transmitted under the maxD/PF/EXP scheduling algorithm in the ITU pedestrian B channel with 16 users per sector.................197

Figure 5.173: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a)-(c) maxD/PF/EXP, and (d)-(f) maxD/PF scheduling algorithms in the ITU pedestrian A channel with 16 users per sector 198

Figure 5.174: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a)-(c) maxD/PF/EXP, and (d)-(f) maxD/PF scheduling algorithms in the ITU pedestrian B channel with 16 users per sector 199

Figure 5.175: Average delay per packet per drop versus number of users per sector for the maxD/PF/EXP scheduling algorithm in the ITU pedestrian A channel 200

Figure 5.176: Average delay per packet per drop versus number of users per sector for the maxD/PF/EXP scheduling algorithm in the ITU pedestrian B channel..............................200

Figure 5.177: Distributions of average packet delay per user for the maxD/PF/EXP scheduling algorithm in ITU pedestrian A channel with 16 users per sector...201

Figure 5.178: Distributions of average packet delay per user for the maxD/PF/EXP scheduling algorithm in ITU pedestrian B channel with 16 users per sector...201

Figure 5.179: Cumulative distribution functions of delays experienced by packets under the maxD/PF/EXP scheduling algorithm in the ITU pedestrian A channel with 16 users per sector 202

Figure 5.180: Cumulative distribution functions of delays experienced by packets under the maxD/PF/EXP scheduling algorithm in the ITU pedestrian B channel with 16 users per sector 203

Figure 5.181: Probability of delays experienced by packets being larger than a given value under the maxD/PF/EXP scheduling algorithm in the ITU pedestrian A channel with 16 users per sector 203

Figure 5.182: Probability of delays experienced by packets being larger than a given value under the maxD/PF/EXP scheduling algorithm in the ITU pedestrian B channel with 16 users per sector 204

Figure 5.183: Distributions of average packet delay per user for the maxD/PF/EXP scheduling algorithm in ITU pedestrian A channel with 16 users per sector 205

Figure 5.184: Distributions of average packet delay per user for the maxD/PF/EXP scheduling algorithm in ITU pedestrian B channel with 16 users per sector 205

Figure 5.185: Distributions of allocated slots per user per drop for the maxD/PF/EXP scheduling algorithm in the ITU pedestrian A channel with 16 users per sector 206

Figure 5.186: Distributions of allocated slots per user per drop for the maxD/PF/EXP scheduling algorithm in the ITU pedestrian B channel with 16 users per sector 206

Figure 5.187: CDF of actual SIR 3 slots in the future (including Rayleigh component and imperfections) for the ITU vehicular B channel, given that the SIR_p (with imperfections, no Rayleigh component) in the current slot supports DRC12 (i.e. SIR_p>9.187 dB) 229

Figure 5.188: Distributions of average packet delay per user for all users (including those whose receive no service for an entire drop) in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins 233

Figure A.1: PER performance in an AWGN channel for an R kb/s transmission of a 1024-bit packet using a maximum of 16 slots (1 of 2) 271

Figure A.2: PER performance in an AWGN channel for an R kb/s transmission of a 1024-bit packet using a maximum of 16 slots (2 of 2) 271

Figure A.3: PER performance in an AWGN channel for an R kb/s transmission of a 1024-bit packet using a maximum of 8 slots 272

Figure A.4: PER performance in an AWGN channel for an R kb/s transmission of a 1024-bit packet using a maximum of s slots 272

Figure A.5: PER performance in an AWGN channel for an R kb/s transmission of a 2048-bit packet using a maximum of s slots 273

Figure A.6: PER performance in an AWGN channel for an R kb/s transmission of a β-bit packet using a maximum of s slots 273

Figure B.1: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian A channel 274

Figure B.2: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian B channel 275

Figure B.3: Distributions of average throughput per user per drop in the ITU pedestrian A channel with 16 users per sector 276

Figure B.4: Distributions of average throughput per user per drop in the ITU pedestrian B channel with 16 users per sector 277

Figure B.5: Distribution of effective bit rates for transmitted packets in the ITU pedestrian A channel with 16 users per sector 278

Figure B.6: Distribution of effective bit rates for transmitted packets in the ITU pedestrian B channel with 16 users per sector 279

Figure B.7 (Part 1 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian A channel with 16 users per sector 280

Figure B.7 (Part 2 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian A channel with 16 users per sector 281

Figure B.8 (Part 1 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian B channel with 16 users per sector 282

Figure B.8 (Part 2 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian B channel with 16 users per sector 283

Figure B.9: Average delay per packet per drop versus number of users per sector in the ITU pedestrian A channel 284

Figure B.10: Average delay per packet per drop versus number of users per sector in the ITU pedestrian B channel 285

Figure B.11: Average packet delay per user per drop versus number of users per sector in the ITU pedestrian A channel on a linear and a logarithmic scale 286

Figure B.12: Average packet delay per user per drop versus number of users per sector in the ITU pedestrian B channel on a linear and a logarithmic scale 287

Figure B.13: Cumulative distribution functions of delays experienced by packets in the ITU pedestrian A channel with 16 users per sector 288

Figure B.14: Cumulative distribution functions of delays experienced by packets in the ITU pedestrian B channel with 16 users per sector 289

Figure B.15: Probability of delays experienced by packets being larger than a given value in the ITU pedestrian A channel with 16 users per sector 290

Figure B.16: Probability of delays experienced by packets being larger than a given value in the ITU pedestrian B channel with 16 users per sector 291

Figure B.17: Distributions of average packet delay per user for users who receive at least one packet in the ITU pedestrian A channel with 16 users per sector..292

Figure B.18: Distributions of average packet delay per user for users who receive at least on packet in the ITU pedestrian B channel with 16 users per sector..293

Figure B.19: Distributions of allocated slots per user per drop in the ITU pedestrian A channel with 16 users per sector 294

Figure B.20: Distributions of allocated slots per user per drop in the ITU pedestrian B channel with 16 users per sector 295

Figure C.1: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian A channel using perfect multislot prediction...296

Figure C.2: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian B channel using perfect multislot prediction...297

Figure C.3: Distributions of average throughput per user per drop in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction 298

Figure C.4: Distributions of average throughput per user per drop in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction 299

Figure C.5 (Part 1 of 2): Distribution of effective bit rates for transmitted packets in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction...............300

Figure C.5 (Part 2 of 2): Distribution of effective bit rates for transmitted packets in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction...............301

Figure C.6 (Part 1 of 2): Distribution of effective bit rates for transmitted packets in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction...............302

Figure C.6 (Part 2 of 2): Distribution of effective bit rates for transmitted packets in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction...............303

Figure C.7 (Part 1 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction 304

Figure C.7 (Part 2 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction 305

Figure C.8 (Part 1 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction 306

Figure C.8 (Part 2 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction 307

Figure C.9: Average delay per packet per drop versus number of users per sector in the ITU pedestrian A channel using perfect multislot prediction... 308

Figure C.10: Average delay per packet per drop versus number of users per sector in the ITU pedestrian B channel using perfect multislot prediction... 309

Figure C.11: Average packet delay per user per drop versus number of users per sector in the ITU pedestrian A channel on a linear and a logarithmic scale using perfect multislot prediction 310

Figure C.12: Average packet delay per user per drop versus number of users per sector in the ITU pedestrian B channel on a linear and a logarithmic scale using perfect multislot prediction 311

Figure C.13: Cumulative distribution functions of delays experienced by packets in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction............... 312

Figure C.14: Cumulative distribution functions of delays experienced by packets in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction............... 313

Figure C.15: Probability of delays experienced by packets being larger than a given value in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction 314

Figure C.16: Probability of delays experienced by packets being larger than a given value in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction 315

Figure C.17: Distributions of average packet delay per user for users who receive at least one packet in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction.. 316

Figure C.18: Distributions of average packet delay per user for users who receive at least one packet in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction.. 317

Figure C.19: Distributions of allocated slots per user per drop in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction 318

Figure C.20: Distributions of allocated slots per user per drop in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction 319

Figure D.1: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian A channel using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 320

Figure D.2: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian B channel using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 321

Figure D.3: Distributions of average throughput per user per drop in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 322

Figure D.4: Distributions of average throughput per user per drop in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 323

Figure D.5: Distribution of effective bit rates for transmitted packets in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 324

Figure D.6: Distribution of effective bit rates for transmitted packets in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 325

Figure D.7 (Part 1 of 3): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 326

Figure D.7 (Part 2 of 3): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 327

Figure D.7 (Part 3 of 3): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 328

Figure D.8 (Part 1 of 3): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 329

Figure D.8 (Part 2 of 3): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 330

Figure D.8 (Part 3 of 3): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 331

Figure D.9: Average delay per packet per drop versus number of users per sector in the ITU pedestrian A channel using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 332

Figure D.10: Average delay per packet per drop versus number of users per sector in the ITU pedestrian B channel using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 333

Figure D.11: Average packet delay per user per drop versus number of users per sector in the ITU pedestrian A channel using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 334

Figure D.12: Average packet delay per user per drop versus number of users per sector in the ITU pedestrian B channel using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 335

Figure D.13: Cumulative distribution functions of delays experienced by packets in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 336

Figure D.14: Cumulative distribution functions of delays experienced by packets in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 337

Figure D.15: Probability of delays experienced by packets being larger than a given value in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 338

Figure D.16: Probability of delays experienced by packets being larger than a given value in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 339

Figure D.17: Distributions of average packet delay per user for users who receive at least one packet in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 340

Figure D.18: Distributions of average packet delay per user for users who receive at least one packet in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 341

Figure D.19: Distributions of allocated slots per user per drop in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 342

Figure D.20: Distributions of allocated slots per user per drop in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 343

Figure E.1: Average sector throughput per drop vs. number of users per sector in the ITU vehicular B channel using outdated SIR_p values and margins 344

Figure E.2: Distributions of average throughput per user per drop in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins 345

Figure E.3 (Part 1 of 2): Distribution of effective bit rates for transmitted packets in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins 346

Figure E.3 (Part 2 of 2): Distribution of effective bit rates for transmitted packets in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins 347

Figure E.4 (Part 1 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins 348

Figure E.4 (Part 2 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins 349

Figure E.5: Average delay per packet per drop versus number of users per sector in the ITU vehicular B channel using outdated SIR_p values and margins 350

Figure E.6: Average packet delay per user per drop versus number of users per sector in the ITU vehicular B channel on a linear and a logarithmic scale using outdated SIR_p values and margins 351

Figure E.7: Cumulative distribution functions of delays experienced by packets in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins 352

Figure E.8: Probability of delays experienced by packets being larger than a given value in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins 353

Figure E.9: Distributions of average packet delay per user for users who receive at least one packet in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins 354

Figure E.10: Distributions of allocated slots per user per drop in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins 355

Figure F.1: Average sector throughput per drop vs. number of users per sector in the ITU vehicular B channel using outdated SIR_p values and margins and comparing the "PF A" and "PF B" average rate calculation methods 356

Figure F.2: Distributions of average throughput per user per drop in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins and comparing the "PF A" and "PF B" average rate calculation methods 357

Figure F.3 (Part 1 of 2): Distribution of effective bit rates for transmitted packets in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins and comparing the "PF A" and "PF B" average rate calculation methods 358

Figure F.3 (Part 2 of 2): Distribution of effective bit rates for transmitted packets in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins and comparing the "PF A" and "PF B" average rate calculation methods 359

Figure F.4 (Part 1 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins and comparing the "PF A" and "PF B" average rate calculation methods 360

Figure F.4 (Part 2 of 3): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins and comparing the "PF A" and "PF B" average rate calculation methods 361

Figure F.4 (Part 3 of 3): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins and comparing the "PF A" and "PF B" average rate calculation methods 362

Figure F.5: Average delay per packet per drop versus number of users per sector in the ITU vehicular B channel using outdated SIR_p values and margins and comparing the "PF A" and "PF B" average rate calculation methods 363

Figure F.6: Average packet delay per user per drop versus number of users per sector in the ITU vehicular B channel on a linear and a logarithmic scale using outdated SIR_p values and margins and comparing the "PF A" and "PF B" average rate calculation methods 364

Figure F.7: Cumulative distribution functions of delays experienced by packets in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins and comparing the "PF A" and "PF B" average rate calculation methods 365

Figure F.8: Probability of delays experienced by packets being larger than a given value in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins and comparing the "PF A" and "PF B" average rate calculation methods 366

Figure F.9: Distributions of average packet delay per user for users who receive at least one packet in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins and comparing the "PF A" and "PF B" average rate calculation methods 367

Figure F.10: Distributions of allocated slots per user per drop in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins and comparing the "PF A" and "PF B" average rate calculation methods 368

Figure G.1: Average sector throughput per drop vs. number of users per sector in the ITU vehicular B channel using perfect one-slot prediction and margins 369

Figure G.2: Distributions of average throughput per user per drop in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins 370

Figure G.3 (Part 1 of 2): Distribution of effective bit rates for transmitted packets in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins 371

Figure G.3 (Part 2 of 2): Distribution of effective bit rates for transmitted packets in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins 372

Figure G.4 (Part 1 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins 373

Figure G.4 (Part 2 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins 374

Figure G.5: Average delay per packet per drop versus number of users per sector in the ITU vehicular B channel using perfect one-slot prediction and margins 375

Figure G.6: Average packet delay per user per drop versus number of users per sector in the ITU vehicular B channel on a linear and a logarithmic scale using perfect one-slot prediction and margins 376

Figure G.7: Cumulative distribution functions of delays experienced by packets in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins 377

Figure G.8: Probability of delays experienced by packets being larger than a given value in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins 378

Figure G.9: Distributions of average packet delay per user for users who receive at least one packet in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins 379

Figure G.10: Distributions of allocated slots per user per drop in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins 380

Figure H.1: Average sector throughput per drop vs. number of users per sector in the ITU vehicular B channel using perfect one-slot prediction and margins, comparing the "PF A" and "PF B" average rate calculation methods 381

Figure H.2: Distributions of average throughput per user per drop in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins, comparing the "PF A" and "PF B" average rate calculation methods 382

Figure H.3: Distribution of effective bit rates for transmitted packets in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins, comparing the "PF A" and "PF B" average rate calculation methods 383

Figure H.4: Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins, comparing the "PF A" and "PF B" average rate calculation methods 384

Figure H.5: Average delay per packet per drop versus number of users per sector in the ITU vehicular B channel using perfect one-slot prediction and margins, comparing the "PF A" and "PF B" average rate calculation methods 385

Figure H.6: Average packet delay per user per drop versus number of users per sector in the ITU vehicular B channel using perfect one-slot prediction and margins, comparing the "PF A" and "PF B" average rate calculation methods 386

Figure H.7: Cumulative distribution functions of delays experienced by packets in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins, comparing the "PF A" and "PF B" average rate calculation methods 387

Figure H.8: Probability of delays experienced by packets being larger than a given value in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins and comparing the "PF A" and "PF B" average rate calculation methods 388

Figure H.9: Distributions of average packet delay per user for users who receive at least one packet in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins, comparing the "PF A" and "PF B" average rate calculation methods 389

Figure H.10: Distributions of allocated slots per user per drop in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins, comparing the "PF A" and "PF B" average rate calculation methods 390

Figure I.1: Average sector throughput per drop vs. number of users per sector in the ITU vehicular B channel using perfect multislot prediction 391

Figure I.2: Distributions of average throughput per user per drop in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction 392

Figure I.3 (Part 1 of 2): Distribution of effective bit rates for transmitted packets in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction....................393

Figure I.3 (Part 2 of 2): Distribution of effective bit rates for transmitted packets in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction....................394

Figure I.4 (Part 1 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction 395

Figure I.4 (Part 2 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction 396

Figure I.5: Average delay per packet per drop versus number of users per sector in the ITU vehicular B channel using perfect multislot prediction 397

Figure I.6: Average packet delay per user per drop versus number of users per sector in the ITU vehicular B channel on a linear and a logarithmic scale using perfect multislot prediction ...398

Figure I.7: Cumulative distribution functions of delays experienced by packets in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction.................399

Figure I.8: Probability of delays experienced by packets being larger than a given value in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction.................400

Figure I.9: Distributions of average packet delay per user for users who receive at least one packet in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction401

Figure I.10: Distributions of allocated slots per user per drop in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction 402

Figure J.1: Average sector throughput per drop vs. number of users per sector in the ITU vehicular B channel using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 403

Figure J.2: Distributions of average throughput per user per drop in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 404

Figure J.3: Distribution of effective bit rates for transmitted packets in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods..405

Figure J.4 (Part 1 of 3): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 406

Figure J.4 (Part 2 of 3): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 407

Figure J.4 (Part 3 of 3): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 408

Figure J.5: Average delay per packet per drop versus number of users per sector in the ITU vehicular B channel using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 409

Figure J.6: Average packet delay per user per drop versus number of users per sector in the ITU vehicular B channel using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 410

Figure J.7: Cumulative distribution functions of delays experienced by packets in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 411

Figure J.8: Probability of delays experienced by packets being larger than a given value in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 412

Figure J.9: Distributions of average packet delay per user for users who receive at least one packet in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 413

Figure J.10: Distributions of allocated slots per user per drop in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods 414

Figure K.1: An infinitesimal ring section within a hexagonal sector 415

Figure K.2: Three cases of the angle θ as a function of the radius r 416

Figure K.3: PDF and CDF distributions of a user being located a distance r from the vertex of a hexagonally-shaped sector, given that the user is located randomly and uniformly over the area of the sector 419

Figure K.4: Probability distributions of at least n users out of N total per sector being located at a distance larger than a radius r from the vertex of the sector 420

Figure K.5: Upper percentiles for probability distributions of at least n users out of N total per sector being located at a distance larger than a radius r from the vertex of the sector 421

List of Symbols

$\mathbf{1}_k(t)$	Equal to 1 if user k is allocated a slot/new packet at time t and zero otherwise in the slot-fair/packet-fair algorithms, respectively
A_{tot}	The total area of a sector
C_k	The minimum throughput user k should receive under the minperf algorithm
DRC_k	Requested rate of user k
$DRC^k_{\max}$	Maximum possible data rate for the packet scheduled for user k
E_S/N_0	The ratio of symbol energy to noise power spectral density; equivalent to the signal-to-noise ratio in an AWGN channel
$E_S/N_{0(1\%)}$	The required signal-to-noise ratio to obtain a 1% packet error rate in the AWGN channel for a given format
$F[k]$	Filter sequence to create a Rayleigh random process with an appropriate auto-correlation and spectrum
G_b	Base station antenna bore sight gain
I_k	Total interference power from non-serving sectors for user k
I_k^x	Interference power of finger x for user k
$J_0(\cdot)$	Zero-order Bessel function of the first kind
L_c	Antenna cable loss
L_{msd}	Multi-screen diffraction loss in the COST231-Walfish-Ikegami path loss model
L_o	Free-space loss in the COST231-Walfish-Ikegami path loss model
L_{ori}	Orientation loss in the COST231-Walfish-Ikegami path loss model
L_p	Total path loss in the COST231-Walfish-Ikegami path loss model
L_{p_max}	Maximum allowable path loss
L_{pen}	Penetration loss

L_{rts}	Roof-top-to-street diffraction and scatter loss in the COST231-Walfish-Ikegami path loss model
$N_{\max}$	The maximum number of re-transmission attempts for a given transmission format
$\bar{P}_{cap\_k}^{xi}$	Power captured by finger x from sector i for user k
P_{fx}	Power captured by finger x in a RAKE receiver
P_{rec}	Received power
$\bar{P}_{rec}^{i}$	Average received signal power from sector i including path loss, antenna pattern, and shadow fading
$P_{rec\_k}^{xi}$	Received power by finger x from sector i for user k
$\bar{P}_{rec\_k}^{xi}$	Average received power including path loss, antenna pattern, and shadow fading by finger x from sector i for user k
P_{rec_min}	Minimum required received power at mobile
P_t	Effective data transmission power
P_{t_max}	Total transmission power per sector
$\bar{R}_{av}(t)$	The mean value of the average rates across all users requesting service at time t
R_{denom}	A constant intended to normalize the maxD metric to the range of the PF metric in an early version of the maxD/PF algorithm
R_k^i	A Rayleigh distributed variable
$R_k(t)$	"PF A" method: equal to the requested rate of user k in slot t if user k is scheduled in slot t, equal to zero otherwise; "PF B" method: equal to the maximum possible data rate of the packet scheduled at time t if a new packet transmission started for user k in slot t, equal to zero if a re-transmission is sent to user k in slot t or if user k is not scheduled in slot t
$\bar{R}_k(t)$	Average rate of user k at time t
$R_x(\tau)$	Auto-correlation of the CDMA spreading sequence

$S(f)$	Power spectral density of the real and imaginary parts of a Doppler-shifted received signal
$\widehat{SIR}$	Signal-to-interference ratio used to determine the supportable format for a mobile
SIR_I	Signal power to imperfection-interference power
SIR_I^T	Total signal power to imperfection-interference power
SIR_k	Signal-to-interference ratio of user k
SIR_M	Signal-to-interference ratio modified to account for system imperfections
$\mathrm{SIR_p}$	Signal-to-interference ratio, not including Rayleigh component of signal
T_C	Coherence time of the fading channel
T_c	Chip period
$U_k(t)$	The "value" or worth of transmitting a packet to user k at time t in the slot-fair, packet-fair, and minperf algorithms
$W_k(t)$	The head-of-line packet delay for user k at time t
X^i	Shadow fading process for sector i
Z, Z^i	Zero-mean, identically distributed Gaussian processes with standard deviation σ_Z used to create cross-correlation in the shadow fading processes between cells
a	Tuning parameter for the maxD/PF and maxD/PF/EXP algorithms; a parameter that controls how quickly the algorithm adapts to changes in the allocated proportions or average rates for the slot-fair/packet-fair and minperf algorithms, respectively
b	Building separation in the COST231-Walfish-Ikegami path loss model
b_k	The weight of the head-of-line delay for user k
$\overline{bW}$	The average of the weighted head-of-line delays across all users receiving service
c	The speed of light (approximately 3×10^8 m/s)
d_{dec}	Decorrelation distance for shadow fading; the distance for which the auto-correlation of the shadow fading drops to 50%
f_c	Carrier frequency of the wireless system

f_m	Maximum Doppler frequency of the mobile
f_s	Sampling frequency
h_b	Height of the base station antennas in the COST231-Walfish-Ikegami path loss model
h_m	Height of the mobile in the COST231-Walfish-Ikegami path loss model
h_{Roof}	Average height of buildings in the COST231-Walfish-Ikegami path loss model
k_m	A parameter used in the filter $F[k]$
p_k	The proportion of slots/packets user k should receive in the long run under the slot-fair/packet-fair algorithms, respectively
r	A Rayleigh distributed random variable, or the radius from the vertex of a sector
$r_k(t)$	The channel capacity of user k at time t; equivalent to DRC_k in the 1xEV-DO system
t_c	Time constant for PF algorithm, controls window size over which rates are averaged
t_s	Sampling period
v	Velocity of the mobile
$v_k(t)$	A parameter that keeps track of the proportion of resources allocated to user k at time t in the slot-fair and packet-fair algorithms
w	Width of streets in the COST231-Walfish-Ikegami path loss model
$\Delta_{1\%}$	A margin tuned so as to provide an average packet error rate of 1%
$\Delta_{\mathrm{DRC}x}$	A margin tuned so as to provide an average packet error rate of 1% for format DRCx
$\Phi(\cdot)$	Auto-correlation of shadow fading process
Ω_p	Total received envelope power
α	Cross-correlation of shadow fading between cells; a parameter used to modify signal-to-interference ratios to account for system imperfections; an angle used in some calculations
$\alpha_k(t)$	A parameter that keeps track of if user k is receiving its minimum specified throughput at time t under the minperf algorithm
β	An angle used in various calculations

γ	An exponentially distributed random variable with mean $\bar{\gamma}$
$\bar{\gamma}$	The mean of the exponentially distributed random variable γ
γ_k	A multiplier in the M-LWDF and EXP algorithms
θ	An angle used in various calculations
ξ	A parameter used to control the shadow fading spatial auto-correlation
σ_X	Standard deviation of shadow fading process
σ_Z	Standard deviation of the processes Z^i
σ_υ	Standard deviation of process υ_k
υ_k	A discrete-time, zero-mean white Gaussian process with standard deviation σ_υ
$\phi(\cdot)$	Auto-correlation of the real and imaginary parts of the complex envelope of a received signal

List of Acronyms and Abbreviations

1xEV-DO	The EVolution of the 1x, or 1.25 MHz, version of cdma2000 designed for Data Only
1xEV-DV	The EVolution of the 1x, or 1.25 MHz, version of cdma2000 designed for Data and Voice
2G	Second Generation
2.5G	Second-and-a-half Generation
3G	Third Generation
8-PSK	8-ary Phase Shift Keying
16-QAM	16-ary Quadrature Amplitude Modulation
ACK	Acknowledgement
ARQ	Automatic Repeat reQuest
AWGN	Additive White Gaussian Noise
CDF	Cumulative Distribution Function
CDMA	Code Division Multiple Access
C/I	Carrier-to-Interference Ratio
COST	European Co-Operation in the field of Scientific and Technical research
CQI	Channel Quality Indicator
DRC	Data Rate Control
EXP	Exponential Rule
HDR	High Data Rate
HOL	Head-of-Line
HRPD	High Rate Packet Data
HSDPA	High Speed Downlink Packet Access

IDFFT	Inverse Discrete Fast Fourier Transform
ITU	International Telecommunication Union
MAC	Medium Access Control
MAP	Maximum *a posteriori*
M-LWDF	Modified Largest Weighted Delay First
NAK	Negative Acknowledgement
NLoS	Non-Line-of-Sight
PDF	Probability Density Function
PER	Packet Error Rate
PF	Proportionally Fair
"PF A"	Average rate calculation method that does not take into account early terminations
"PF B"	Average rate calculation method that accounts for early terminations
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RF	Radio Frequency
RR	Round Robin
RRI	Reverse Rate Indicator
SNR	Signal-to-Noise Ratio
SIR	Signal-to-Interference Ratio
UMTS	Universal Mobile Telecommunications System
WCDMA	Wideband Code Division Multiple Access
maxD	Maximum DRC / Maximum Rate

maxD/PF	Maximum Rate / Proportionally Fair hybrid
maxD/PF/EXP	Maximum Rate / Proportionally Fair / Exponential Rule hybrid
minperf	Minimum Performance
packet-fair	Packet-wise Fair
slot-fair	Slot-wise Fair

Chapter 1: Introduction

1.1. Background and Motivation

The very first generation of cellular radio systems was analog and designed almost exclusively to handle voice traffic. Later, as second generation (2G) systems were developed, the switch was made to digital schemes. The emphasis of these 2G systems was still voice traffic. That is, circuit-switched connections were used, with each user being allocated a certain channel for communications. That user maintained continuous use of the channel for the entire duration of a call. With the advent of digital service, however, there was rudimentary support for packet data. The data rates that were available were rather small, though. For example, in IS-95 [1], a CDMA-based (code division multiple access) system, the maximum data rate was 9.6 kb/s, although that maximum was increased later to 115.2 kb/s with the addition of supplemental code channels to the standard. These data rates were sufficient for simple services, such as receiving short text messages, but were too low for anything beyond that.

Later cellular systems provided a bit more emphasis on packet data. (Such systems are generally referred to as second-and-a-half generation, or 2.5G systems.) Somewhat higher data rates became available. For example, in the earlier releases of IS-2000 [2] (also known as cdma2000), a maximum data rate of 307.2 kb/s was available. However, due to issues such as system loading and interference, the actual data rates achieved by users were generally well below that maximum. As such, although certain services like text-based e-mail and web browsing became more common, the low data rates prevented any more substantial activities.

As cellular systems evolve into the third generation (3G) and beyond, packet data has now become the main focus in the design of the systems. The goal of these systems is to enable high throughput packet data on the forward link to users. Most of the traffic on the Internet today is asymmetrical in nature, with much more data being received by users from a source (on the forward link) than is sent from those users (on the reverse link). Much of that data is tolerant to delay; that is, the packets can experience delays on the order of seconds, and the resulting service (e.g. e-mail, web browsing) will not really be affected. This tolerance to delay can be exploited in order to increase the throughput of the system.

Several techniques have been developed in order to facilitate the high bit rates desired for 3G systems. To begin, system resources are no longer divided up among several users on a circuit-switched basis. In several papers ([3],[4],[5]), it has been shown that the capacity of a system with a single transmit and receive antenna and in a fading environment is highest when only a single user receives data at a time. That is, the system should not attempt to serve several users at a time, but rather at any given time, the transmitter should allocate all its power to transmitting data to a single user. Multiple users then share the resources by dividing up the system usage into short time slots, with one user receiving service in each time slot. Each time slot is on the order of one or two milliseconds. The short length of the slot helps to reduce packet errors, as the less time spent transmitting a packet, the smaller the likelihood of the channel conditions changing during that slot.

In earlier generations of cellular systems, techniques such as handoff and power control were employed for the circuit-switched connections. Handoff is a technique that handles the movement of the mobile between cells and between sectors of the same cell. The signal strength of the mobile is monitored constantly by several sectors/cells. As the mobile user moves within a region, the connection is transferred to the transmitter that sees the best signal strength from the user. In CDMA systems, this transfer can be done without breaking the connection and reassigning the user to a different channel. Power control was used to manage the interference within a sector. The mobiles raise and lower the power of their transmitted signals continually so that the received power from each of the mobiles at the base station is approximately the same. These two techniques are no longer required for the forward link of evolutionary cellular systems. Because the system usage is divided into time slots, with only one user served in each slot, intra-sector interference is effectively eliminated. Power control is not used; rather, the base station transmits at full power during each of the slots. Instead of handoff, the mobile constantly monitors the received signal strength from several sectors. The mobile then requests packets from the sector with the strongest signal.

Another element important in enabling high throughput packet data transmissions is the use of adaptive modulation and coding. As mentioned above, the transmitter does not change its power level; it transmits at full power in each slot. Rather than changing the power, the system instead adapts the packet size, data rate, error correction coding, and modulation scheme for each user according to the maximum rate the user who is receiving data can support. The proper transmission format is known at the transmitter due to feedback regarding the channel conditions from the users. This could take the form of a requested rate and format, or simply a raw measurement of the signal-to-interference ratio (SIR) as estimated at the receiver.

Yet another technique to increase throughput is the use of ARQ (Automatic Repeat reQuest). The simplest form of ARQ is just an error detection and re-transmission scheme. When a packet error is detected, the mobile informs the transmitter, which then re-transmits that packet. There are three commonly used types of re-transmission schemes. Under stop-and-wait, under which the transmitter waits for a response from the mobile regarding if the packet was correct or in error before transmitting the next packet. With ARQ, the receiver attempts to decode the received packet. When using the go-back-N scheme, the transmitter sends packets continuously until the mobile informs the base station of an error. At that point, the transmitter goes back N packets in its queue and re-transmits those N packets to the user. The selective repeat scheme is similar to the go-back-N scheme, except the transmitter only re-sends the one particular packet that was in error.

To further increase the throughput of the system, an error correction code can be added to the packets to reduce the likelihood of packet errors. This combination of error correction and detection is known as hybrid ARQ. If no additional processing is done to the packets, the system is referred to as Type I hybrid ARQ. However, the system throughput can be increased even further through the use of incremental redundancy and soft packet combining. Because of the error correction coding, it is possible to correctly decode the packet before the entire packet is received. Hence, for the first transmission of a packet, the transmitter could just send the information bits of the packet, plus a few parity bits to assist in the decoding process. If re-transmissions are required, the transmitter can instead just send more parity bits from the packet to help with the decoding, thereby increasing the effective strength of the error correction code. Alternately, some previously transmitted bits can be re-sent. In either case, the receiver can soft combine the various transmissions in order to obtain a more reliable copy of the packet. This combination of error notification, re-transmissions, and incremental redundancy is known as Type II hybrid ARQ. In the event that the packet can be decoded early (e.g. before the entire packet is sent, or before the maximum number of re-transmissions are used), the effective bit rate of that packet is increased, thereby further raising the throughput of that user and the system overall. With Type II hybrid ARQ, the stop-and-wait re-transmission scheme is most commonly used due to its low complexity. Note however that this does not mean that a user cannot receive more than one packet at once, nor that only a packet for one particular user can be handled at a time. Packets can be interlaced by using multiple channels, or, in the context of the methods already discussed, by interlacing different packets across the various time slots. During the slots that the transmitter waits for a response for a given packet, other packets may be transmitted. These may be delivered to other users, or to the same user from whom the transmitter is waiting a response.

One final way of increasing the system throughput is through the use of scheduling. Because only one user receives a packet during any given time slot, a scheduling algorithm is required to determine which user receives data in each slot. The throughput of the system can thereby be increased by making use of multiuser diversity. When there are several users requesting packets in a given sector, some of the users will be in good channel conditions, while others will be in poorer conditions. Because much of the data transmitted on the forward channel is tolerant to delays, the scheduler can postpone transmissions to the users in the poorer conditions until the conditions improve and those users can support higher bit rates. By transmitting to users in the relatively best conditions at any given time, the throughput of the system in the fading channel can be increased relative to what can be achieved in a non-fading, static channel. The scheduler must also attempt to meet several conflicting goals. It is desirable to maximize the total throughput of the system, so as to allow the operator of the system to make full use of the available resources and increase the resulting revenue the system generates. The scheduler must also ensure some measure of fairness to all users requesting service. Other factors may also play a role in the scheduling decision. For example, depending on the type of data a user is requesting, there may be certain Quality of Service (QoS) parameters that must be satisfied. These may include elements such as guaranteeing each user a certain minimum throughput, or ensuring that the delay experienced by packets does not exceed a certain value. Hence, the scheduling algorithm can have a major impact on the overall performance of the system. The design and analysis of these scheduling algorithms is the major focus of this work.

1.2. Objectives and Overview of Thesis

The main goal of this thesis is to analyze the performance of various scheduling algorithms in a high throughput packet data cellular radio system. Specifically, the impact of the scheduling algorithm on the throughputs achieved by the system and the users in the system, on the delays experienced by packets and by users, and on the fairness of the system to the various users is of interest. Some of the algorithms that are examined are pre-existing in the literature. Others are modifications of existing algorithms, and some are newly proposed in this work. The performance of all these algorithms is evaluated through a series of computer simulations.

Of a secondary nature, it was mentioned earlier that the mobiles estimate their signal-to-interference ratios, and from that, the bit rate and format that each of the mobiles can support is determined. In an actual system, there is a delay between the time the signal strength is measured and the point where a transmission begins to a user. This delay is on the order of a few slots. As a consequence, some kind of prediction of the signal in future slots is desirable. However, the subject of prediction is quite vast and beyond the scope of this work. Instead, two methods are used to determine the formats the users can support. In the first, outdated channel information is used along with margins tuned to obtain a 1% packet error rate per format used. That is, the margins are tuned such that approximately 1% of the packets sent under each transmission format are in error. The second format determination method simply assumes that the mobiles have perfect knowledge of what their SIR values will be in future slots, and hence know exactly the formats they can support. The way in which the mobiles decide upon the transmission format they can support affects the decisions made by the scheduling algorithm and hence the performance of the overall system. Consequently, the effect of the format determination method on the performance of the scheduling algorithms was also examined.

The remainder of this thesis is organized as follows. Chapter 2 discusses some of the details of two existing high throughput packet data systems. Chapter 3 outlines the various scheduling algorithms analyzed in this work. Chapter 4 describes the system model and simulation structure used to perform the evaluations of the scheduling algorithms. The results of the simulations for the various algorithms, format determination methods, and average rate calculation methods are presented in Chapter 5. Finally, a summary and conclusions as well as possible future work are provided in Chapter 6.

Chapter 2: High Throughput Downlink Packet Data Access Systems

2.1. Introduction

In second generation wireless systems, such as IS-95 [1] and the earlier releases of IS-2000 [2] (also known as cdma2000), the main emphasis of the system design was on voice transmissions. There was some capability of data transmission as well, but only at relatively low bit rates (i.e. a maximum of 9.6 kb/s for IS-95 or 115.2 kb/s with the later addition of supplemental code channels, and 307.2 kb/s for IS-2000, although the rates achieved in practice are typically much lower due to system loading and interference from multiple users). These rates are sufficient to provide simple data services, such as sending and receiving text-based messages and web pages. However, the rates are not high enough to support activities such as receiving web pages or e-mail with graphics. Hence, systems that can support higher bit rates were required.

There have been several systems proposed and standardized to enable high bit rate packet data on the forward link (from the base station to the mobile). Some of these systems include IS-856 [6] and 1xEV-DV as described in Release C of IS-2000 [7] for the evolution of North American cdma2000, and HSDPA [8],[9] for Euro-Japanese WCDMA (also known under its European acronym UMTS, or Universal Mobile Telecommunications System). These systems are different in terms of the specific details of their designs. However, they all share common traits in order to enable high bit rates over the wireless channel.

One such trait is the use of very short time slots (on the order of one or two milliseconds) to transmit packets. This helps to mitigate fading effects, as the coherence time of the channel would be much longer than the length of the time slot, and so the channel would appear constant during the transmission of a single packet.

It should be noted that the trend for these high throughput systems is to use spread-spectrum techniques as the basis for the transmitted signal. The transmitted signal is broadened over a wide frequency range with an orthogonal spreading code, and then is later de-spread at the receiver. The spreading cover displays an extremely low auto-correlation (i.e. the inverse of the number of chips used in the pseudo-noise spreading sequence) if two copies of the signal are offset in time by one chip or more. A useful result of this low auto-correlation is that multiple copies of the transmitted signal that arrive at the receiver due to a multipath environment can be distinguished from each other if the time difference between the paths is at least one chip long. The multiple paths can each be captured in a RAKE receiver and combined to obtain a better copy of the received signal. This technique can effectively convert a dispersive channel into a non-dispersive one.

Power control on a cell's downlink is no longer employed. It has been shown in several papers ([3], [4], [5]) that the highest system capacity is achieved when the base station always transmits at full power, and all of that power is directed towards a single user. System usage is divided into time slots, with a single user in a sector receiving data in any given slot. This method helps to reduce interference, as there are no longer intra-sector signals to other users to contend with. The only interference now comes from other sectors. Because all the transmit power for the data channel is directed towards a single user, adaptive modulation and coding are used to maximize the data rate that user can receive under the given channel conditions. The mobiles perform estimation and prediction of their received signal-to-interference ratios (SIRs) as often as every slot. They then feed back to the base station in some form or another information regarding what kind of channel conditions they are experiencing (based on the SIR estimation and prediction) in order to enable this adaptation. Best sector selection is also employed instead of handoff. The mobile monitors the received signal strength from several sectors and requests data from the sector with the strongest signal.

To further increase the effective bit rates, stop-and-wait type II hybrid ARQ (Automatic Repeat reQuest) is used. Since channel conditions can be better than predicted, it is possible to correctly decode the packet prior to the maximum number of re-transmissions being reached. After each transmission attempt is received, the mobile attempts to decode the entire packet. The mobile uses an acknowledgement channel to inform the base station if decoding was successful or not after each attempt. This allows the base station to determine whether a re-transmission for the existing packet should be sent to that user, or if a new packet can be started instead. Thus, the actual achieved data rate of that packet can be higher than the nominal data rate, which occurs when the maximum number of slots or re-transmissions are used. When re-transmissions are required, the system can employ incremental redundancy. That is, the data sent during re-transmissions can contain more parity bits to help decode the packet, or they can contain copies of bits sent during earlier transmission attempts, or a combination of both. The transmission attempts are soft combined together in this process.

The final trait that the various systems have in common is scheduling of packet transmissions in order to exploit multi-user diversity. By postponing transmission to a user for a short period of time until said user has a relatively good channel, the net throughput of the system and of the users will increase. The assumption and emphasis of this work is on such delay-tolerant packet data.

The next two sections describe two high bit rate packet data systems for the evolution of the cdma2000 system. The descriptions focus on the forward link; the reverse link is only described where it relates to the operation of the forward link, such as in the case of feedback channels. Only the relevant details are provided; for a full description, the standard documents themselves are referenced.

2.2. 1xEV-DO

The 1xEV-DO (Data-Only Evolution of the 1x, or 1.25 MHz, version of cdma2000) system is also known by the names High Data Rate (HDR) [10] and

High Rate Packet Data (HRPD), and has been standardized as IS-856 [6]. (The name HDR more accurately refers to an earlier pre-standardization version of the system, in which hybrid ARQ was not employed.) The system is optimized for data service, and operates on a separate RF carrier in the same frequency bands as cdma2000.

On the forward link, there is a Pilot Channel, a Medium Access Control (MAC) Channel, a Traffic Channel and a Control Channel that are time-division multiplexed together in one slot. The reverse link consists of a Pilot Channel, a Data Channel, a Reverse Rate Indicator (RRI) Channel (for the data channel), a Data Rate Control (DRC) Channel, and an Acknowledgement (ACK) Channel. Of these reverse channels, the most important in terms of forward link data traffic are the DRC and ACK channels.

The general structure of a forward link slot (each of length 1.666… ms) is shown below.

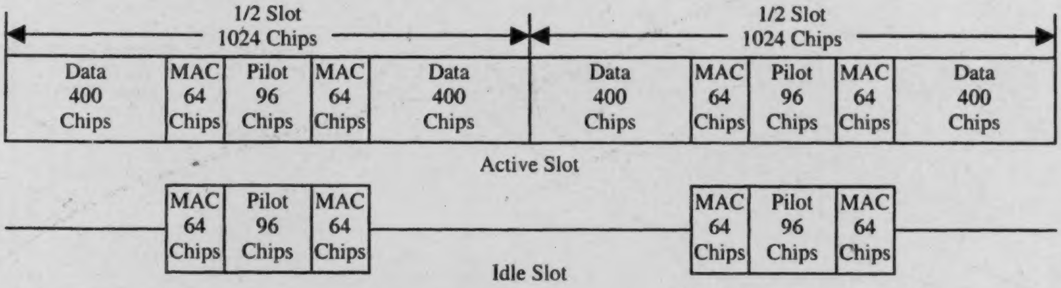

1/2 Slot
1024 Chips
1/2 Slot
1024 Chips
Data 400 Chips
MAC 64 Chips
Pilot 96 Chips
MAC 64 Chips
Data 400 Chips
Data 400 Chips
MAC 64 Chips
Pilot 96 Chips
MAC 64 Chips
Data 400 Chips
Active Slot
MAC 64 Chips
Pilot 96 Chips
MAC 64 Chips
MAC 64 Chips
Pilot 96 Chips
MAC 64 Chips
Idle Slot

Figure 2.1: 1xEV-DO forward link slot structure

On the Forward Traffic Channel, data packets can be sent using 12 possible formats. These formats are outlined in the table below.

TABLE I:
1XEV-DO TRAFFIC CHANNEL TRANSMISSION FORMATS

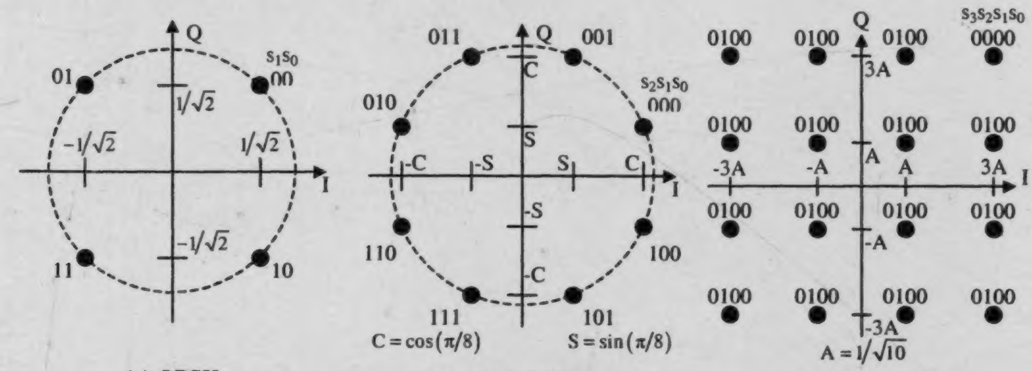
Format	Nominal Data Rate (kb/s)	Max. # of slots (transmissions of a given packet)	Packet Size (bits)	Turbo Code Rate	Modulation Type	# of Symbols (after Turbo Coding and Modulation)	Preamble Length (chips)
1	38.4	16	1024	1/5	QPSK	2560	1024
2	76.8	8	1024	1/5	QPSK	2560	512
3	153.6	4	1024	1/5	QPSK	2560	256
4	307.2	2	1024	1/5	QPSK	2560	128
5	614.4	1	1024	1/3	QPSK	1536	64
6	307.2	4	2048	1/3	QPSK	3072	128
7	614.4	2	2048	1/3	QPSK	3072	64
8	1228.8	1	2048	1/3	QPSK	3072	64
9	921.6	2	3072	1/3	8-PSK	3072	64
10	1843.2	1	3072	1/3	8-PSK	3072	64
11	1228.8	2	4096	1/3	16-QAM	3072	64
12	2457.6	1	4096	1/3	16-QAM	3072	64

Note that the data rates as given in the table above only apply if the maximum number of slots is used to transmit the packet. If fewer slots are used, the actual data rate will be higher. This is discussed in more detail later.

It should also be noted that the actual physical layer packet with size described above consists of a combination of MAC layer packet data bits, padding, parity bits, and a tail sequence. However, the convention when expressing the physical layer data rates is to consider all these bits, regardless of whether they are used by upper layers or not. Thus, the data rates as stated in this paper are in reference to the full physical layer packets.

The error detection and correction coding of the packets is performed in a systematic fashion using a turbo encoder. The output of the encoder is the initial sequence of information bits as fed to the input of the encoder, followed by a sequence of parity bits. For an encoder rate of 1/*R*, there will be *R* times as many output symbols in total than there were input bits. For example, for a rate 1/5 encoder, the first 1/5 of the output symbols will be information bits, while the remaining 4/5 of the output symbols will be parity bits.

When modulating the turbo-coded symbols, it is assumed that the coded and interleaved symbols come in time in the order $s_0s_1s_2s_3s_4\ldots$ and so on. These symbols are then grouped in sets of 2, 3, or 4 (for QPSK, 8-PSK, and 16-QAM respectively) in reverse order. For example, for QPSK, successive groups would be $\{s_1s_0\}$, $\{s_3s_2\}$, $\{s_5s_4\}$, etc., and for 8-PSK, successive groups would be $\{s_2s_1s_0\}$, $\{s_5s_4s_3\}$, etc. These groups are then mapped to the in-phase (I) and quadrature (Q) channels as shown in the signal constellations below. In general, the higher the signal-to-interference ratio a mobile sees from the transmitter, the higher the order of modulation that user will be able to support.



(a) QPSK (b) 8-PSK (c) 16-QAM

Figure 2.2: Signal constellations for (a) QPSK, (b) 8-PSK, and (c) 16-QAM modulations

On both the I and Q channels, the modulation symbols are demultiplexed into 16 parallel streams. Each of those streams is then spread by a different Walsh code of length 16. The resulting sequences are then scaled by a factor of ¼ and summed on a chip-by-chip basis. The scaling ensures that the summed sequence has the same power as the original sequence. It can therefore be seen that there is effectively one modulation symbol transmitted per chip interval.

In order to assist in the synchronization of packet transmissions, in the first slot, prior to data being sent, the data chips are preceded by a preamble. The number of preamble chips depends on the rate of the packet transmission; this number is shown in Table I. These chips appear in the beginning of the "data" portion of the first time slot. The spread data modulation symbols take up the remaining portion of the data chips. If multiple slots are required to transmit a packet, the data sequence continues from where it left off in the previous slot. If the end of the data sequence is reached in the middle of a slot, the sequence simply starts over from the beginning. Therefore, the incremental redundancy of the system can be seen. Only pieces of the packet are sent during each re-transmission, and this process repeats until sufficient data has been sent to decode the packet correctly, or until the maximum number of re-transmissions has been reached. Also note that in some cases, the entire sequence of the turbo-coded symbols does not need to be. The symbols containing the information bits can fit completely within the first slot of a packet transmission, so it is possible with all formats for the data to be decoded correctly after only one slot, without all the parity bits having been sent. In fact, for the highest set of data rates, the entire sequence of parity bits is never sent. The following diagrams demonstrate the multiplexing of the preamble, pilot, MAC, and data chips.

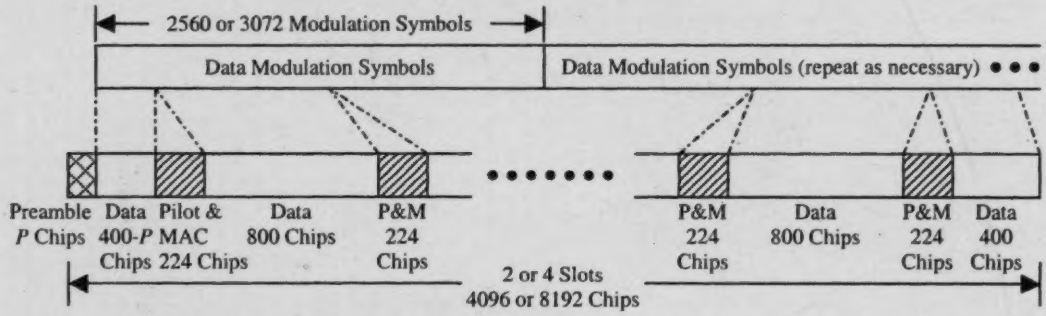

2560 or 3072 Modulation Symbols
Data Modulation Symbols
Data Modulation Symbols (repeat as necessary) • • •
Preamble P Chips
Data 400-P Chips
Pilot & MAC 224 Chips
Data 800 Chips
P&M 224 Chips
P&M 224 Chips
Data 800 Chips
P&M 224 Chips
Data 400 Chips
2 or 4 Slots
4096 or 8192 Chips

Figure 2.3: Preamble, pilot, MAC, and data multiplexing for the multiple-slot cases with data rates of 153.6, 307.2, 614.4, 921.6, and 1228.8 kb/s

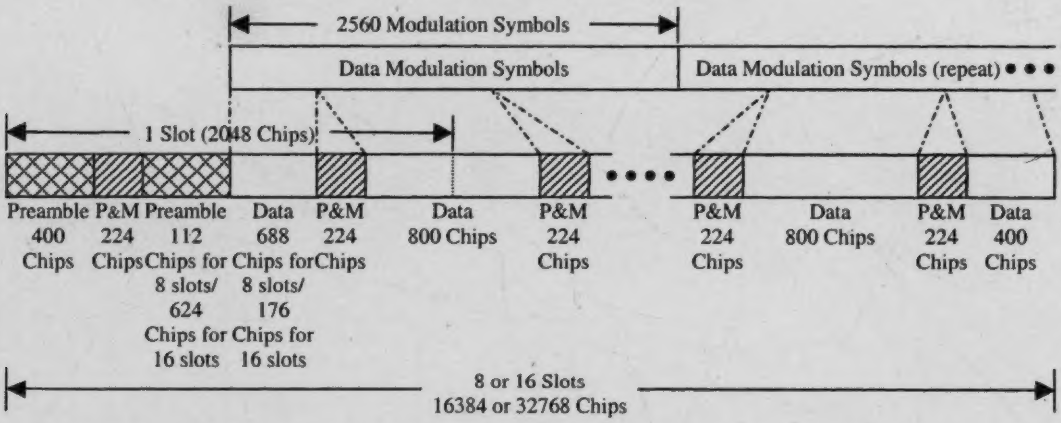

2560 Modulation Symbols
Data Modulation Symbols
Data Modulation Symbols (repeat) • • •
1 Slot (2048 Chips)
Preamble 400 Chips
P&M 224 Chips
Preamble 112 Chips for 8 slots/ 624 Chips for 16 slots
Data 688 Chips for 8 slots/ 176 Chips for 16 slots
P&M 224 Chips
Data 800 Chips
P&M 224 Chips
P&M 224 Chips
Data 800 Chips
P&M 224 Chips
Data 400 Chips
8 or 16 Slots
16384 or 32768 Chips

Figure 2.4: Preamble, pilot, MAC, and data multiplexing with data rates of 38.4 and 76.8 kb/s

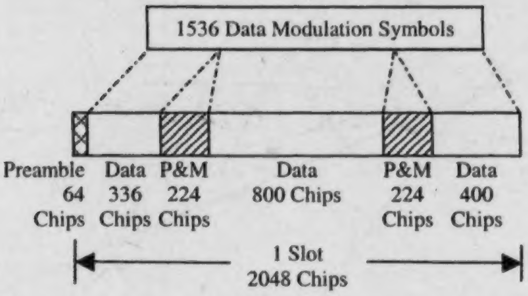

1536 Data Modulation Symbols
Preamble 64 Chips
Data 336 Chips
P&M 224 Chips
Data 800 Chips
P&M 224 Chips
Data 400 Chips
1 Slot
2048 Chips

Figure 2.5: Preamble, pilot, MAC, and data multiplexing for the 1-slot case with a data rate of 614.4 kb/s

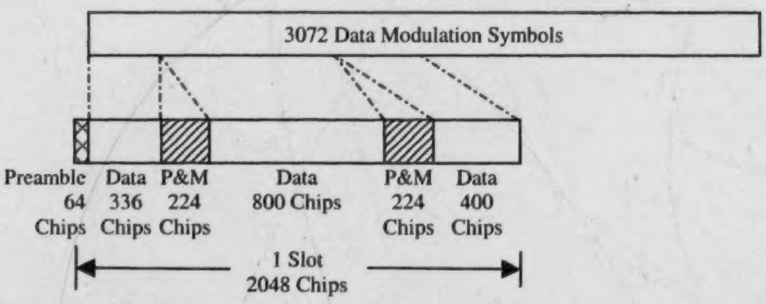

3072 Data Modulation Symbols
Preamble 64 Chips
Data 336 Chips
P&M 224 Chips
Data 800 Chips
P&M 224 Chips
Data 400 Chips
1 Slot
2048 Chips

Figure 2.6: Preamble, pilot, MAC, and data multiplexing for the 1-slot cases with data rates of 1228.8, 1843.2, and 2457.6 kb/s

On the reverse link, as mentioned earlier, the two most important channels are the DRC Channel and the ACK Channel. The DRC Channel is synchronized a half-slot out of step with the Forward Traffic Channel, while the ACK Channel is synchronized with the Forward Traffic Channel. However, transmissions on the ACK channel only occur during the first half of any given slot. The channels are not time-division multiplexed as on the forward link, but rather are separated in code space using different spreading codes.

Each mobile uses the pilot chips from the forward link to estimate and predict the SIR that the mobile would see during the course of a possible transmission. The mobile then uses that value of SIR to determine what transmission format can be supported. (Note that the method of mapping SIRs to formats is not specified by the standard and is left up to the mobile designer. Examples of possible methods can be found in [11] and [12].) A request for that format is made on the DRC Channel to the best serving sector. Which sector the request is made to can be determined by the spreading code used on the DRC Channel. Each sector that the mobile is in communication with and could possibly receive a packet from is assigned one of 8 possible Walsh spreading covers. When sending the DRC request, the mobile uses the cover that corresponds to the sector that the mobile wishes the packet to come from. The base station decides which user to transmit data to in slot n based on the scheduling algorithm that is implemented.

Re-transmissions in 1xEV-DO are synchronous. That is, if re-transmissions of a packet are necessary, they will always occur 4 slots apart.

With those slots reserved for re-transmissions, no new packets can be scheduled at those times until the attempts at transmitting the previous packet have been completed. During the third slot after a transmission attempt is made, the mobile responds on the ACK channel with a positive acknowledgement (ACK or '0') if the packet could be decoded correctly, or a negative acknowledgement (NAK or '1') if the packet could not be decoded correctly. If a NAK is received, a re-transmission is scheduled for the next slot, unless the maximum number of transmission attempts has already been reached. In this latter case, the packet is discarded and a packet error is declared. Since the transmission on the ACK channel occurs during the first half of the slot, the base station still has the second half of the slot available for the scheduling algorithm to decide whom to schedule a new packet to in the next slot if a positive acknowledgment was received. The following two figures show two possible scenarios for a packet transmission using the 153.6 kb/s format.

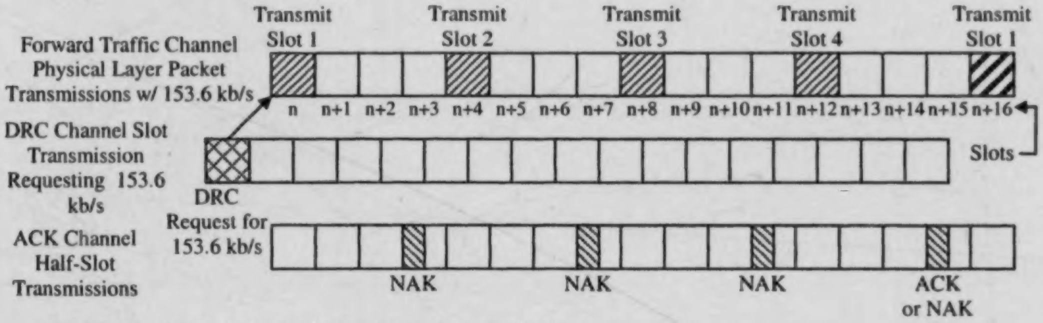

Transmit Slot 1
Transmit Slot 2
Transmit Slot 3
Transmit Slot 4
Transmit Slot 1
Forward Traffic Channel Physical Layer Packet Transmissions w/ 153.6 kb/s
n n+1 n+2 n+3 n+4 n+5 n+6 n+7 n+8 n+9 n+10 n+11 n+12 n+13 n+14 n+15 n+16
Slots
DRC Channel Slot Transmission Requesting 153.6 kb/s
DRC Request for 153.6 kb/s
ACK Channel Half-Slot Transmissions
NAK
NAK
NAK
ACK or NAK

Figure 2.7: Multiple-slot packet with normal termination

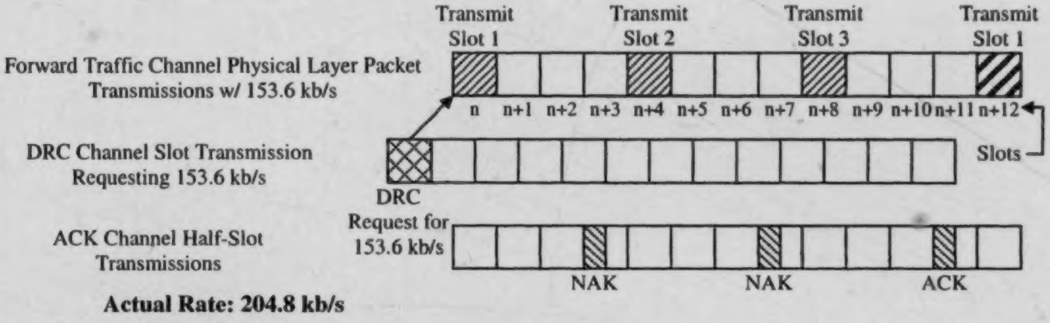

Transmit Slot 1
Transmit Slot 2
Transmit Slot 3
Transmit Slot 1
Forward Traffic Channel Physical Layer Packet Transmissions w/ 153.6 kb/s
n n+1 n+2 n+3 n+4 n+5 n+6 n+7 n+8 n+9 n+10 n+11 n+12
Slots
DRC Channel Slot Transmission Requesting 153.6 kb/s
DRC Request for 153.6 kb/s
ACK Channel Half-Slot Transmissions
NAK
NAK
ACK
Actual Rate: 204.8 kb/s

Figure 2.8: Multiple-slot packet with early termination

Note that in the case of early termination, the packet does not take the maximum number of slots to transmit, and consequently the effective data rate for that packet is higher than nominal. In the above case of a 1024-bit packet over 3 slots, the effective data rate is 204.8 kb/s (in contrast to 153.6 kb/s had the transmission taken 4 slots). Also note that in the interleaved slots (i.e. n+1, n+2, n+3, n+5, etc.) and in the first slot after the previous transmission terminates (slot n+16 in the normal termination example or slot n+12 in the early termination example above), other packets are transmitted. These may be packets to other users, or more packets to the same user. The scheduling algorithm would determine which user receives a packet during which slot. This interlacing of packets increases the throughput of the overall system.

Interference can be reduced by gating transmissions on the DRC channel or by transmitting a given request over multiple slots at a lower power. The reduced interference can increase the capacity of the reverse link. However, due to the increased likelihood of the DRC request being out of date because of changing channel conditions, the throughput on the forward link will drop. Thus, there can be a tradeoff between forward and reverse link capacity. For the simulations, the DRC requests were made on a slot-by-slot basis in order to maximize the forward link capacity.

As a final note, if it were required for Control Channel data to be transmitted on the forward link, that data would take the place of Forward Traffic Channel data in the slot. The transmissions would occur at either 38.4 or 76.8 kb/s, with the same modulation and coding formats as if it were ordinary data traffic. The MAC bits distinguish the Control Channel data from ordinary data. Control Channel traffic would reduce the effective throughput of the system, as slots that would be used to transmit Forward Traffic Channel data to the mobile are instead used for signaling purposes. The simulations ignore the effect of Control Channel data and assume that all packets are intended as Forward Traffic Channel data.

2.3. 1xEV-DV

The 1xEV-DV system is embedded within Release C of the IS-2000 standard [7] as Radio Configuration 10. Since it is part of IS-2000, and as the "DV" part of the name implies, the system is designed to carry both data and voice traffic. Similar to 1xEV-DO, the 1xEV-DV system enables high throughput packet data service by only transmitting to one user at a time on the forward link, with rates, formats, and re-transmissions determined from feedback on the reverse link. However, there are many differences in how the system is implemented.

To begin, on the forward link, there is a Pilot Channel, a Packet Data Channel, and two Packet Data Control Channels. These are no longer time-division multiplexed as in the 1xEV-DO system, but rather are code-division multiplexed. The pilot signal is used by the mobile to estimate the channel quality, while the Packet Data Control Channel is used to indicate to which mobile the transmission belongs and what format the transmission is encoded in. As implied by the previous statement, the mobiles no longer determine for themselves what format they can support. Instead, they only report their raw carrier-to-interference (C/I) values (as estimated from the pilot) to the base station on a reverse link Channel Quality Indicator Channel. There is also an Acknowledgement Channel to report if decoding of packets was successful or not. The slot length is now 1.25 ms in comparison to 1.666… ms for 1xEV-DO.

Adaptive modulation and coding is again a key feature of this system. As mentioned above, one key difference is that the mobiles no longer determine for themselves what packet format they can support. Rather, the base station's scheduling algorithm makes the decision based on the (quantized) C/I values reported by each mobile. The mapping of the C/I values to supportable formats is not specified in the standard, and is left to the designer of the base station. Because raw C/I values can be used in the scheduling decisions, there is now some further soft information about which user is relatively best to transmit to. This additional information could be used to break a tie between two users when making scheduling decisions.

For the possible formats, each encoder packet in the physical layer will contain either 408, 792, 1560, 2328, 3096, or 3864 bits. These bits consist of information bits from the MAC layer, 16 parity bits (a frame quality indicator sequence), and 6 tail bits. The encoder packets are encoded in a rate 1/5 turbo encoder, interleaved, and scrambled. The scrambled bits are then split into four subpackets. The number of bits in a subpacket varies, but is always larger than the size of the original encoder packet. Therefore, the first subpacket of the four will always contain the entire sequence of information bits from the turbo encoder (as well as some parity bits), and hence the entire encoder packet can potentially be recovered from just the first subpacket.

Each subpacket is spread using a combination of anywhere from 1 to 28 Walsh spreading codes of length 32 in order to be transmitted. The subpacket is demultiplexed into parallel streams, then each of the streams is spread and the resulting sequences are summed on a chip-by-chip basis, similar to the method seen for the 1xEV-DO system. The key difference is that the number of spreading codes is not fixed at 16. In general, the more Walsh codes assigned to a subpacket, the smaller the C/I required to receive that subpacket correctly. However, the use of more codes also increases the complexity of encoding and decoding the packet. Effectively, the use of more codes is equivalent to using multiple channels to send a subpacket.

Subpackets can be transmitted using any of the three modulation schemes as discussed in the 1xEV-DO system. (The mapping of groups of symbols into points on the modulation constellations is identical to that described for the 1xEV-DO system.) However, in a difference from 1xEV-DO, the transmission of a subpacket can take place over either 1, 2, or 4 consecutive slots. As with the use of more Walsh codes, using more slots also generally reduces the C/I required to receive the subpacket correctly. However, the subpacket data rate also drops in inverse proportion to the number of slots used.

Overall, the size of the subpacket depends on the number of Walsh codes, the type of modulation scheme, and the number of consecutive slots used to transmit the subpacket. The possible data rates and range of subpacket sizes for the 6 sizes of encoder packets are shown in the table below.

TABLE II:
POSSIBLE ENCODER PACKET SIZES, SUBPACKET DATA RATES, AND RANGE OF SUBPACKET SIZES FOR THE 1xEV-DV SYSTEM

Number of Encoder Packet Bits	Possible Subpacket Data Rates (kb/s)	Range of Possible Subpacket Sizes (bits)

Number of Encoder Packet Bits	Possible Subpacket Data Rates (kb/s)	Range of Possible Subpacket Sizes (bits)
3864	3091.2 (1 slot TX) 1545.6 (2 slot TX) 772.8 (4 slot TX)	4992 - 7296
3096	2476.8 (1 slot TX) 1238.4 (2 slot TX) 619.2 (4 slot TX)	4032 - 7680
2328	1862.4 (1 slot TX) 931.2 (2 slot TX) 465.6 (4 slot TX)	3072 - 7680
1560	1248.0 (1 slot TX) 624.0 (2 slot TX) 312.0 (4 slot TX)	2112 - 5376
792	633.6 (1 slot TX) 316.8 (2 slot TX) 158.4 (4 slot TX)	1056 - 4224
408	326.4 (1 slot TX) 163.2 (2 slot TX) 81.6 (4 slot TX)	576 - 1824

The scheduling algorithm in the MAC layer would control the number of encoder packet bits and the format used for each subpacket. The scheduling algorithm also has complete control over the timing and format of any re-transmissions of subpackets, if necessary. This is a major difference in comparison to the 1xEV-DO system – re-transmissions of subpackets can be asynchronous. That is, it is no longer required for the re-transmission of subpackets to occur 4 slots after the previous transmission (although the designer of the scheduling algorithm may incorporate synchronous re-transmissions as part of the algorithm if it is so desired). Furthermore, the re-transmissions can be any of the four subpackets in any order and in any transmission format. The only stipulations are that the initial subpacket transmitted must be the first of the four possible subpackets (i.e. the one that contains all the information bits from the turbo encoder), and no more than 8 total subpacket transmissions for any given encoder packet are allowed. These differences mean that the scheduling algorithm can delay the re-transmission of subpackets until channel conditions are more favorable for a user, or it can use a different format for a re-transmission that the current channel conditions dictate. In general, this asynchronous re-transmission scheme can lead to higher system throughputs by waiting for better channel conditions, but it can also lead to higher delays for the packets of users with less favorable channels. Asynchronous re-transmission can also lead to increased delay jitter (i.e. the variation in delays seen by the set of packets received by a user), which could be a factor in the performance of certain real-time data transfer applications.

Because the format for each subpacket is decided at the base station instead of the mobile, it is necessary to signal to the users who the current transmission on the Packet Data Channel is intended for and what format is being used to send the subpacket. The Packet Data Control Channels are used for this purpose. The Control Channels also relay such information as which encoder packet is being sent, the size of the encoder packet, and which one of the four possible subpackets is being transmitted for the given encoder packet. The number of slots for the control channel transmission is the same as that for the subpacket concurrently being sent on the Packet Data Channel.

As with the 1xEV-DO system, interleaving of packets is supported, so that a given mobile can receive more than one packet at the same time, without being forced to wait until the previous packets are decoded and the positive or negative acknowledgement is sent to the base station. Up to 4 encoder packets for a given mobile can be handled simultaneously. A difference from the 1xEV-DO system is that when sending the ACKs or NAKs on the Acknowledgement Channel, it is no longer specifically required that the transmissions occur in the third slot after a packet is received. Instead, there is a tunable parameter that specifies the number of slots between the completion of a transmission of a subpacket on the Packet Data Channel and the start of the corresponding transmission of an ACK or NAK on the Acknowledgement Channel. The value of this parameter is specified by one of the upper signaling layers, and the tuning of this parameter can potentially have an impact on the system performance. Also note that transmissions on the Acknowledgement Channel take a full slot, rather than a half-slot as in 1xEV-DO. This means that re-transmissions could not take place the slot immediately following the reception of a negative acknowledgment. Since the re-transmission scheme is no longer synchronous, this does really causes a problem, other than the minor increase in the delay that packet experiences.

On the Reverse Channel Quality Indicator (CQI) Channel, the mobile sends quantized values of its estimated C/I to the base station. There are two types of values that can be sent: absolute and differential. With absolute values, a full report is made on the C/I value. These reports represent the estimated C/I value, quantized from –16.25 dB to 6.25 dB in steps of 1.5 dB. With differential values, only an "UP" or "DOWN" bit is sent, which tells the base station to revise the current C/I estimate by ±0.5 dB, while staying within the range of –16.25 to 6.25 dB. The differential reports have the advantages that they allow a finer tuning of the reported C/I estimate, as well as requiring less power to transmit. The Reverse CQI channel can also operate in a gated mode or by using repetitions of absolute/differential values to further reduce power consumption, thereby increasing battery life at the mobile, reducing interference, and increasing the reverse link capacity. Also note that the sector the mobile wishes to receive data from is incorporated into Reverse CQI Channel based on the spreading code used for the reports. As mentioned earlier, since the raw C/I values are transmitted to the base station instead of requested formats as in 1xEV-DO, there is additional soft information available to the base station. The scheduling algorithm could use this information when deciding which user to transmit a packet to and what format that packet should take. There would also be a savings in complexity and battery power at the mobile, as the device would no longer have to run a signal strength-to-supportable format mapping subroutine.

As a final note, it is possible for a base station to support two Forward Packet Data Channels simultaneously (hence the requirement of two Packet Data

Control Channels). Each channel can use anywhere from 1 to 28 Walsh codes to transmit subpackets, but no more than 28 Walsh codes total can be used between the two channels. Subpackets must be transmitted simultaneously on both channels, with the formats used on either channel at any given time taking the same number of slots. The subpacket transmitted on each channel must be intended for a different mobile. This is a difference from the 1xEV-DO system, and can potentially decrease the delays seen within a system. The decreased delay would likely come with a penalty in reduced sector throughput, however, as the system is splitting its resources between two users.

2.4. Conclusion

This chapter has described two existing high throughput downlink packet data systems for the evolution of cdma2000. However, only the performance of scheduling algorithms as applied to the 1xEV-DO system has been analyzed. The analysis of the algorithms in the one system should provide a good overall view of how well they work in relation to each other when applied to other systems.

Chapter 3: Description of Scheduling Algorithms

3.1. Introduction

As seen in the previous chapter, in the 1xEV-DO system, only one user receives data on the forward traffic channel during any given time slot. The scheduling algorithm determines the selection of that user. This selection process can have a huge effect on the system performance in terms of data throughput, delays experienced by the various packets being transmitted and the users requesting service, the number of time slots allocated to each user, and so on. In general, the scheduling algorithm must attempt to meet two conflicting goals. First, the network operator would desire to have the largest possible amount of data transmitted through the system in order to maximize the revenue that is created. This can be done rather simply by always transmitting to the user with the highest requested data rate. However, this scheme would also lead to a large degree of unfairness amongst the users requesting service. Those users with poorer channel conditions (for example, further away from the transmitter, or perhaps with a large number of scattering objects between the transmitter and the receiver) would receive very little or possibly no data at all. This would lead to a high degree of dissatisfaction from those users, which could possibly result in those users seeking service from another provider. Therefore, some degree of fairness must be assured to all users requesting service. How well the balance between throughput and fairness is achieved is dependent on the scheduling algorithm that is implemented in the system.

There are many scheduling algorithms available for wireline data services, and the throughput, delay, and fairness characteristics of those algorithms under wireline conditions have been well studied. However, there is a vastly different set of conditions when considering a wireless channel. In general, the wireline channel is highly dependable, with very low error rates. Several users often share the bandwidth of the wireline channel simultaneously. The channel capacity is also well known and does not change over time. This is not the case with the wireless channel. Due to the motion of the mobile and the various objects in the surrounding environment, the channel conditions change with time, often quite rapidly. Thus, the capacity of the channel is no longer known beforehand, and must be estimated by signal measurements. This time-varying nature of the channel leads to much higher error rates, and as a result the scheduling algorithms for wireline networks cannot be used as is.

There have been several attempts to modify existing wireline scheduling algorithms to the wireless channel. A good overview of these methods can be found in [13]. Unfortunately, most of these methods are of little use with the 1xEV-DO system. In general, the techniques involved in the adaptation of the existing algorithms involve a "good/bad" model of the wireless channel. That is, either the wireless channel is "good", meaning that a user can receive data, or it is "bad", meaning that no data should be sent to a giver user at that time. However, due to the technique of adaptive modulation and coding, this model is not useful.

Instead of a two-state channel, there are in fact multiple states, depending on the signal-to-interference ratio (SIR). As the SIR experienced by a user increases, the bit rate that can be supported by that user also increases. Any scheduling algorithm used with 1xEV-DO should take this fact into account.

The following sections describe the scheduling algorithms analyzed in this thesis.

3.2. Round Robin (RR)

The Round Robin (RR) algorithm is one of the simplest and most well known algorithms available. In this algorithm, users are simply selected in cyclic order to receive data. In other words, in slot 1, the algorithm would select user 1 to receive data. In the next available slot, the algorithm would select user 2, and so on until user N is scheduled, where N is the number of users requesting service. The algorithm would then start over again at user 1. Note however that some slots may be reserved ahead of time for re-transmission of packets. In this case, the re-transmission takes priority. (This is true for all scheduling algorithms, not just round robin). The next available slot (i.e. that is not reserved for a re-transmission) would go to whoever is next in line in the cyclic order. If a user is in outage and not able to support any data rate, that user is skipped.

This algorithm is one of the fairest algorithms, in that all users receive approximately the same number of packets. However, the algorithm also makes no use whatsoever of multiuser diversity. As mentioned earlier, the 1xEV-DO system is optimized for delay-tolerant data traffic. By selecting users with relatively good channel conditions to receive data and delaying users with relatively poorer channel conditions until their channel improves, there will be a substantial increase in the amount of throughput that can be achieved by the system. Since the RR algorithm does not take into account channel conditions in the scheduling decision, the throughput achieved by RR will be lower than by an algorithm that takes channel conditions into account. (This assumes, of course, that the estimation of the channel conditions is reasonably accurate. If the estimates are not accurate, then the knowledge of which user is relatively the best is flawed, and the system performance deteriorates.) Consequently, the RR algorithm provides a lower bound on the system throughput.

3.3. Maximum Rate / Maximum DRC (maxD)

At the other end of the scale from Round Robin is the Maximum Rate or Maximum DRC (Data Rate Control) algorithm (maxD). This algorithm always selects for transmission the user k with the highest requested rate DRC_k, with ties broken randomly. The maxD algorithm is similar in nature to the more commonly known Max C/I (carrier-to-interference ratio) algorithm. However, it is not quite the same. Consider two users with similar SIRs (i.e. within about 1-2 dB). Both users could wind up requesting the same data rate. However, the base station does not know which user has the better signal between the two; only the requested rate is known. Therefore, since the tie is broken at random, the user with the lower SIR could be selected.

The maxD algorithm provides the highest possible system throughput, and hence is the upper bound. However, it achieves this throughput in a very unfair manner. Since only the user(s) with the best channel conditions is/are selected, the remaining users would receive very little service, or possibly no service at all. Thus, there would be a large variation in terms of the amount of throughput received and the delay experienced across the set of users. Hence, the maxD algorithm is the least fair of all the algorithms analyzed.

3.4. Proportionally Fair (PF)

The concept of proportional fairness has been described by Frank Kelly in [14]. A system is proportionally fair for a set S of rates x_s if, by changing to a new set of rates x_s^*, the following holds:

$$\sum_{s \in S} \frac{x_s^* - x_s}{x_s} \leq 0 \tag{3.1}$$

Stated another way, a system is proportionally fair if by increasing the throughput of one user by $x\%$, the sum of the percentages of drops in throughput for the remaining users will be greater than or equal to $x\%$.

"PF A"

The Proportionally Fair algorithm for 1xEV-DO has been studied by David Tse and others (e.g. [15], [16], [17], [18]). The algorithm selects in slot t the user k for transmission with the largest ratio of requested rate to average rate $DRC_k(t)/\bar{R}_k(t)$. Therefore, the user with the largest current supportable rate relative to the average rate seen in the past is chosen to receive data. In this way, the user who is chosen is always near a peak in his signal strength.

The average rate $\bar{R}_k(t)$ for each user is calculated over a window of past transmission slots using a first-order lowpass filter as follows:

$$\bar{R}_k(t) = \left(1 - \frac{1}{t_c}\right)\bar{R}_k(t-1) + \frac{1}{t_c} R_k(t-1) \tag{3.2}$$

The value $R_k(t-1)$ is equal to the requested rate of user k if that user was scheduled to receive a transmission or a re-transmission in slot t–1; otherwise, the value is equal to zero if user k was not selected to receive any type of transmission in slot t–1. The constant t_c is the window size of the past number of slots that are considered in the averaging. The value of t_c can affect the performance of the algorithm. If the value is small, only the past few slots are averaged, which in the extreme case means that essentially only the current conditions are being considered. In this event, the algorithm selects the user with the current best rate overall, making the performance equivalent to the maxD algorithm. At the other extreme, a very large value of t_c would mean that any fluctuations due to small-scale and large-scale fading and shadowing would be averaged out. This would mean that no user would be relatively better than any other over the long term, and so the performance of the algorithm would be similar to if users were simply selected in cyclic order as in the RR algorithm, or if they were simply chosen at random. A value of t_c=1000 slots is used in this work, as suggested in [16]. The average rates are updated every slot prior to the scheduling decision being made.

"PF B"

It should be noted that the above method of calculating $\overline{R}_k(t)$ was originally derived using a pre-standard version of 1xEV-DO. In the early stages of the development of the system, 1xEV-DO did not incorporate any form of ARQ. This meant that all packets took the maximum number of re-transmissions to be sent, regardless of whether the packet could be decoded correctly before that maximum was reached. Thus, early terminations are not considered in the above calculation of $\overline{R}_k(t)$, since they were not possible when the formula was first derived. As a consequence, the value of $\overline{R}_k(t)$ can be somewhat low when early terminations occur. Because of this fact, users who experience a large number of early terminations (e.g. those users in a vehicular channel) will be favored by the PF algorithm due to their smaller denominator in the $DRC_k(t)/\overline{R}_k(t)$ ratio.

To account for early terminations, the following change to the PF algorithm is proposed. The method of calculating $\overline{R}_k(t)$ is largely the same as described above. The difference is in how the value of $R_k(t-1)$ is determined:

$$R_k(t-1) = \begin{cases} DRC_{\max}^k, & \text{if a new packet is scheduled for user } k \text{ in slot } t-1 \\ 0, & \text{if a re-transmission is scheduled for user } k \text{ in slot } t-1 \\ 0, & \text{if user } k \text{ is not scheduled to receive a packet in slot } t-1 \end{cases} \quad (3.3)$$

The value $DRC_{\max}^k$ is the maximum possible data rate for the given packet size scheduled for user k in slot t–1; in other words, it is the rate that would be achieved if the packet only takes one slot to receive. For a packet size of 1024 bits (Formats 1 to 5), this is 614.4 kb/s, for 2048 bits (Formats 6 to 8), it is 1228.8 kb/s, for 3072 bits (Formats 9 and 10), it is 1843.2 kb/s, and for 4096 bits (Formats 11 and 12), it is 2457.6 kb/s. By using this method, the early terminations are accounted for. In the event that a transmission does take the maximum possible number of slots, the value of $\overline{R}_k(t)$ that results from either method is approximately the same.

To differentiate between the two $\overline{R}_k(t)$ methods during the performance analysis, the calculation method as used prior to ARQ and early terminations being considered shall be designated "PF A", while the new calculation method proposed here which takes early terminations into account shall be designated "PF B". In the event that the proportionally fair algorithm or a method of calculating $\overline{R}_k(t)$ is referred to without specifying which method, the "PF A" method is implied, as it is the more widely known method in the literature.

3.5. Maximum Rate / Proportionally Fair Hybrid (maxD/PF)

The goal of this algorithm is to create a hybrid of the Maximum Rate and Proportionally Fair algorithms with a tunable parameter to control the throughput and delay performance characteristics between those provided by the maxD and the PF algorithms. Initially, the idea was to select the user k with the largest value of:

$$a\frac{DRC_k(t)}{\bar{R}_k(t)}+(1-a)\frac{DRC_k(t)}{R_{denom}} \tag{3.4}$$

The value a is the tunable parameter, while the value R_{denom} would be a constant that would normalize the value of the maxD metric in the second half of the equation to the range of the PF metric in the first half. Unfortunately, this equation did not yield satisfactory results using the specified constant. More specifically, it was desired that by adjusting the tunable parameter a, an approximately linear change in throughput between that obtained with the maxD algorithm and that obtained from the PF algorithm would result for any number of users. This, unfortunately, did not prove to be the case. It was later determined that the denominator did not necessarily have to be a constant in time, but only a constant across all users. This led to the final usable form of the equation:

$$a\frac{DRC_k(t)}{\bar{R}_k(t)}+(1-a)\frac{DRC_k(t)}{\bar{R}_{av}(t)} \tag{3.5}$$

The value a is still the tunable parameter to control the throughput and delay characteristics, and has a value that can range between 0 to 1 inclusive. When a is zero, the algorithm is equivalent to the maxD algorithm. Similarly, when a=1, the algorithm is equivalent to the PF algorithm. The new denominator $\bar{R}_{av}(t)$ is equal to the average value of $\bar{R}_k(t)$ across the set of users requesting service in slot t. (Either method for calculating $\bar{R}_k(t)$ as described above in the PF algorithm section could be used.)

More details on how the final form of the maxD/PF scheduling metric was derived can be found in the performance analysis chapter.

3.6. Modified Largest Weighted Delay First (M-LWDF)

The authors in [19] describe a family of algorithms that they call Modified Largest Weighted Delay First. In this algorithm, the user k is scheduled for transmission based on the largest value of:

$$\gamma_k W_k(t) r_k(t) \tag{3.6}$$

The value $r_k(t)$ is the channel capacity with respect to user k; in other words, this would be the requested rate in 1xEV-DO. $W_k(t)$ is the head-of-line (HOL) packet delay for user k. For the performance analysis, the HOL delay is defined as the time between a packet arriving at the front of the queue of packets for user k (which occurs once the previous packet in the queue has been scheduled for its first transmission slot) until that packet is scheduled to begin transmission. In this manner, the minimum possible HOL delay for a packet is 1 slot. Say a new packet (e.g. packet number $j-1$) is scheduled for user k in slot $t-1$, so that packet j arrives at the front of the queue. Further, assume that packet j is immediately scheduled for transmission in slot t. (Packet $j+1$ would now be at the head of the queue.) The HOL delay for packet j would be 1 slot.

The value γ_k is a multiplier that may or may not be different for each user, and may or may not be constant. In the event that it is the same for all users and constant, the algorithm is essentially a maximum rate-maximum delay product. This case is studied in the performance analysis as "M-LWDF 1". The authors in [19] also suggest that γ_k could be made proportional to $1/\bar{R}_k(t)$ as in the PF algorithm. In this event, the metric becomes a PF-maximum delay product. This latter case is analyzed as "M-LWDF 2".

3.7. Slot-wise Fair and Packet-wise Fair (slot-fair & packet-fair)

Liu, Chong, and Shroff describe in [20] an algorithm that maximizes the system throughput under the constraint of resource sharing. In the case of this algorithm, the resource that is shared is time; in the long run, each user is allowed to use the system for a specified proportion of the time. In 1xEV-DO, this is equivalent to users being allocated a certain proportion of the overall slots; hence, the algorithm is described as slot-wise fair. The optimization of throughput is accomplished by defining a value or worth $U_k(t)$ of a potential packet transmission to user k. This worth is set to the requested rate of user k, normalized to the lowest possible rate of 38.4 kb/s. In other words, a requested rate of 38.4 kb/s would have a worth of 1, 76.8 kb/s would have a worth of 2, and so on up to 2457.6 kb/s, which would be worth 64. The user k is then selected for transmission based on who has the largest sum of $[U_k(t)+v_k(t)]$, where $v_k(t)$ is a parameter updated every slot according to:

$$v_k(t+1) = v_k(t) - a\left(\mathbf{1}_k(t) - p_k\right) \tag{3.7}$$

The value $\mathbf{1}_k(t)$ equals 1 if user k is scheduled for a transmission or re-transmission in slot t; if user k is not scheduled in slot t, the value of $\mathbf{1}_k(t)$ is zero. p_k is defined as the proportion of the time slots user k should receive in the long run. The value a is a multiplier used to track changes in the channel conditions, system, and number of slots a user receives. The value of a can change with time; however, it was found that using a constant of a=0.25 provided good performance in terms of slot allocation to the various users. This case is analyzed under the identifier of "slot-fair".

Ideally, it would be desirable to directly compare the slot-wise fair algorithm with the RR algorithm. Unfortunately, while the two algorithms do both have resource-sharing goals, the goal of round robin is to share packets, not slots. If all packets took the same number of slots to transmit, then both goals would be equivalent. However, different transmit formats have differing maximum numbers of transmit slots possible, and there are also early terminations to consider. Thus, for a one-to-one comparison with round robin, another packet-sharing algorithm is necessary. Fortunately, it is easy to modify the slot-wise fair algorithm to do just that. Note that the parameter $v_k(t)$ is updated every slot in order to keep track of the number of slots allocated to each user. If instead of updating every slot, the update were done only when a new packet was scheduled for a user, the parameter would keep track of the number of packets allocated to each user instead of the number of slots. This modification can therefore be called a Packet-wise Fair algorithm, and can directly be compared with the RR algorithm. As part of this modification, the performance analysis (done under the name "packet-fair") showed that the value of a should be changed to 0.5 to maintain good packet allocation characteristics. In addition, the packet-wise fair algorithm is also capable of compensating a user if it goes into outage for a period of time, allowing it to catch up in terms of the number of packets received compared to the other users. The RR algorithm does not have this compensation mechanism.

There were various interesting effects seen with these two algorithms in the event that certain users went into outage and were unable to support any data rate for an extended period of time. These effects are discussed in the performance analysis chapter.

3.8. Minimum Performance (minperf)

The minimum performance ("minperf") algorithm was also created by Liu, Chong, and Shroff, and is described in [21]. This algorithm also maximizes the system throughput, this time under the constraint that each user be given (or attempted to be given) a specified minimum throughput C_k. That minimum may be different for all users in the system, although for the analysis, the value of C_k for all users was made identical. The authors also state that the feasibility of that minimum specification should be checked in advance, in order to ensure that the minimum performance is possible in the first place. Unfortunately, it is not particularly feasible to check for feasibility in a multiuser wireless system. The number of users in the system, the amount and type of traffic they request, their individual channel conditions, and the order in which the scheduler winds up selecting the users will all affect the throughput each user winds up receiving, and it is almost impossible to check this in advance. Therefore, rather than checking the feasibility of the minimum rate, this step was ignored. While this might not be advisable in all cases, it does at the very least allow an interesting study of what happens when one attempts to provide all users with a specified throughput when that throughput is not possible.

The minperf algorithm uses the same worth $U_k(t)$ as defined in the slot-fair algorithm. To track the data throughput allocated to each user, a parameter $\alpha_k(t)$ is used, with the user k with the largest product of $\alpha_k(t)U_k(t)$ being chosen for transmission. Both the minimum and the initial value of α_k for all users is 1, and the parameter is updated every slot. Unfortunately, the authors do not specify in

[21] how the update should be calculated. However, based on the method used to update the parameter in the slot-fair algorithm and further information given in [22], the update is accomplished first by calculating $\overline{R}_k(t)$ as in the PF algorithm, then by updating $\alpha_k(t)$ as follows:

$$\alpha_k(t) = \max\left\{\alpha_k(t-1) - a\left[\overline{R}_k(t) - C_k\right], 1\right\} \tag{3.8}$$

The units of $\overline{R}_k(t)$ and C_k are in bits per second, and the value of a used in the performance analysis is 5×10^{-5}. The analysis of this algorithm was conducted for values of C_k ranging from 5000 bits/s to 25000 bits/s in steps of 5000 bits/s. It should also be noted that the maxD algorithm is a special case of this algorithm, when C_k=0 bits/s for all users.

3.9. The Exponential Rule (EXP)

The exponential rule was first described in a Bell Labs technical memo by Shakkottai and Stolyar [23]. It is similar to the M-LWDF algorithm, but involving an exponential term of the head-of-line packet delay (as defined for the M-LWDF algorithm) instead the HOL delay by itself. In the exponential rule, the user k is scheduled for transmission in slot t with the largest value of:

$$\gamma_k r_k(t) \exp\left(\frac{b_k W_k(t)}{1+\sqrt{\overline{bW}}}\right) \tag{3.9}$$

The values of γ_k, $r_k(t)$, and $W_k(t)$ are the same as described in the M-LWDF algorithm. The values b_k are weights for each user (all set to 1 for the simulations), and the $\overline{bW}$ term is the weighted average of the HOL packet delays across all N users requesting service:

$$\overline{bW} = \tfrac{1}{N}\sum_k b_k W_k(t) \tag{3.10}$$

Since γ_k is the same as in the M-LWDF algorithm, the same two cases arise for possible values of γ_k: a constant ("EXP 1"), or proportional to $1/\overline{R}_k(t)$ ("EXP 2"). There is a further complication, however: the algorithm metric will be affected based on whether the delay is expressed in units of slots or in units of seconds. With the M-LWDF algorithm, this was not a problem, since the delay term directly multiplied the $\gamma_k r_k(t)$ part of the metric. If the delay was in slots or in seconds, the metric would be scaled by a constant amount for all users (i.e. $1.666\ldots\times10^{-3}$ seconds/slot). This would not affect which user had the relatively largest metric. However, with the EXP rule, the delay is contained within the exponential term, so the units will affect the overall value, and not in a constant manner across all users. Hence, four cases for the EXP rule are analyzed: for units of delay in slots ("EXP1A" and "EXP2A") and in units of seconds ("EXP1B" and EXP2B").

Note that the exponential term does not become significant until the weighted delay of one of the users becomes on the order of $1+\sqrt{\overline{bW}}$. Until that time, the algorithm acts as either the maxD algorithm in the EXP1 cases or the PF algorithm in the EXP2 cases.

3.10. The maxD/PF/EXP Hybrid

With the EXP rule, until the exponential term becomes significant, the algorithm is essentially the same as the maxD or the PF algorithm. In both of these cases, this is accomplished by multiplying either the maxD metric or the PF metric by the exponential delay term. Consequently, it is trivial to add an EXP rule extension to the maxD/PF hybrid introduced above. All that is necessary is to multiply the maxD/PF metric by the exponential delay term:

$$\left[a\frac{DRC_k(t)}{\overline{R}_k(t)}+(1-a)\frac{DRC_k(t)}{\overline{R}_{av}(t)}\right]\exp\left(\frac{b_k W_k(t)}{1+\sqrt{\overline{bW}}}\right) \qquad (3.11)$$

The terms a, $DRC_k(t)$, $\overline{R}_k(t)$, and $\overline{R}_{av}(t)$ are the same as in the maxD/PF hybrid, and the b_k, $W_k(t)$, and $\overline{bW}$ terms are the same as in the exponential rule algorithm.

Chapter 4: System Model and Simulation Structure Used for Performance Evaluation

4.1. Introduction

Creating a simulation of a wireless communications system is a somewhat complicated process. Besides the requirement of modeling the system of interest, it is also necessary to model the environment in which the system will be operating. In order to simplify this process, the simulation was broken up into three separate parts: a link-level simulation, a cellular radio environment simulation, and a system performance simulation.

The link-level simulation provides an AWGN link performance evaluation of the transmission formats used in the system. Essentially, it determines what the expected packet error rate (PER) will be in an AWGN channel at different E_S/N_0 values if data are sent using the formats specified in the 1xEV-DO standard. The cellular radio environment simulation determines how the signal-to-interference ratios (SIRs), which are equivalent to the E_S/N_0 values used in the AWGN link simulations, change over time in a multi-cell system. Finally, the system performance simulation takes the results of the link-level and radio environment simulations, and combines them with factors such as system imperfections, specific scheduling algorithms, and packet error determinations to obtain overall system performance results. This last stage also collects the final system statistics in terms of throughput, delay, and fairness. These three stages are shown in the figure below, and are described in more detail in the following sections.

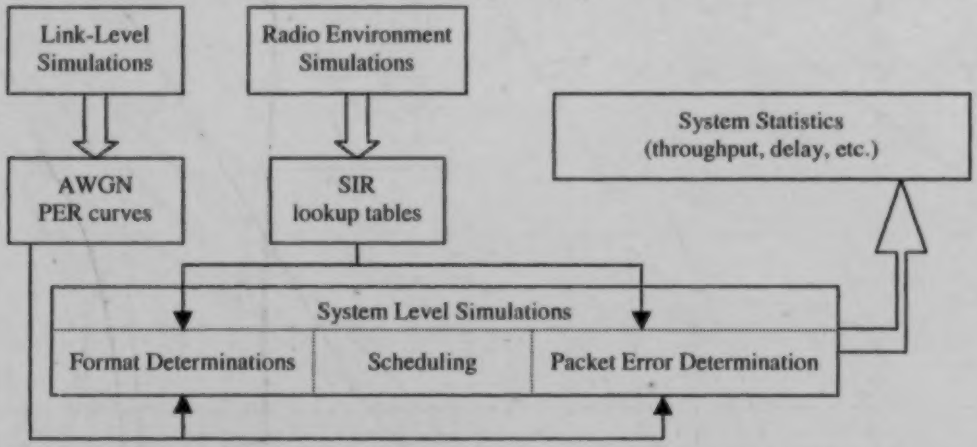

Link-Level Simulations
Radio Environment Simulations
System Statistics (throughput, delay, etc.)
AWGN PER curves
SIR lookup tables
System Level Simulations
Format Determinations
Scheduling
Packet Error Determination

Figure 4.1: Block diagram of simulation process

4.2. Link-Level Simulations

The main purpose of the link-level simulations is to evaluate the performance of the various formats used in 1xEV-DO as described in Chapter 2. This includes varying such parameters as the packet size, turbo coding rate, modulation size, number of slots used to transmit the packet, preamble length, and so on. The desired result is to generate curves of packet error rates (PERs) versus the E_S/N_0 ratio (symbol energy to noise power spectral density) in the additive white Gaussian noise (AWGN) channel.

In practice, the wireless channel changes over time, possibly quite rapidly depending on the carrier frequency and the velocity at which the mobile moves. However, during the duration of a single slot, the channel conditions are approximately constant. This can be demonstrated by calculating the coherence time of the channel, which is defined at the duration over which the amplitudes of received signals will be highly correlated. As given in [24], a popular rule of thumb for calculating the coherence time is to use the following equation:

$$T_C = \sqrt{\frac{9}{16\pi f_m^2}} \approx \frac{0.423142}{f_m} \qquad (4.1)$$

The value f_m is the maximum Doppler shift of the mobile, which is given by $f_m = vf_c/c$, where v is the velocity of the mobile, f_c is the carrier frequency of the system, and c is the speed of light. It is obvious that as the velocity of the mobile increases, the Doppler shift increases proportionally, and hence the coherence time will decrease. We will be considering two types of channels: pedestrian channels in which the mobile is assumed to move at 3 km/h (0.83333... m/s), and a vehicular channel, in which the mobile speed is assumed to be 60 km/h (16.6666... m/s). (These will be discussed in more detail in the radio environment simulation section.) The carrier frequency of the forward link of the 1xEV-DO system is in the range of 1930 to 1990 MHz for the PCS band in North America. Thus, considering the vehicular channel, and using a mid-range frequency of 1960 MHz, the coherence time would be:

$$T_C \approx \frac{0.423142}{f_m} = \frac{0.423142c}{vf_c} \approx \frac{0.423142\left(3\times10^8 \text{ m/s}\right)}{\left(16.6666 \text{ m/s}\right)\left(1960\times10^6 \text{ Hz}\right)} = 3.9 \text{ ms} \qquad (4.2)$$

The coherence time of the vehicular channel of 3.9 ms is much larger than the slot duration of 1.6666... ms, and so the use of the AWGN channel for the link-level simulations is justified. In a pedestrian channel, the coherence time would be 20 times as long, or about 78 ms.

In the 1xEV-DO system, there are 12 different formats, each allowing from 1 to 16 slots maximum for the transmission of a packet. However, it is possible for a packet to be received correctly using any number of slots up to the maximum for the given format. Overall, this works out to 41 possible ways for a packet to be received. These methods are summarized in the table below.

TABLE III:
POSSIBLE RECEIVED PACKET FORMATS FOR 1xEV-DO

Format	Packet Size (bits)	Max. Slots	Slots Used	Effective Data Rate (kb/s)	Effective Code Rate	Modulation Scheme
DRC1	1024	16	16	38.4	1/5	QPSK
DRC1	1024	16	15	40.96	1/5	QPSK
DRC1	1024	16	14	43.89	1/5	QPSK

Format	Packet Size (bits)	Max. Slots	Slots Used	Effective Data Rate (kb/s)	Effective Code Rate	Modulation Scheme
DRC1	1024	16	13	47.26	1/5	QPSK
DRC1	1024	16	12	51.2	1/5	QPSK
DRC1	1024	16	11	55.85	1/5	QPSK
DRC1	1024	16	10	61.44	1/5	QPSK
DRC1	1024	16	9	68.27	1/5	QPSK
DRC1	1024	16	8	76.8	1/5	QPSK
DRC1	1024	16	7	87.77	1/5	QPSK
DRC1	1024	16	6	102.4	1/5	QPSK
DRC1	1024	16	5	122.88	1/5	QPSK
DRC1	1024	16	4	153.6	1/5	QPSK
DRC1	1024	16	3	204.8	1/5	QPSK
DRC1	1024	16	2	307.2	4/17	QPSK
DRC1	1024	16	1	614.4	8/9	QPSK
DRC2	1024	8	8	76.8	1/5	QPSK
DRC2	1024	8	7	87.77	1/5	QPSK
DRC2	1024	8	6	102.4	1/5	QPSK
DRC2	1024	8	5	122.88	1/5	QPSK
DRC2	1024	8	4	153.6	1/5	QPSK
DRC2	1024	8	3	204.8	1/5	QPSK
DRC2	1024	8	2	307.2	1/5	QPSK
DRC2	1024	8	1	614.4	8/17	QPSK
DRC3	1024	4	4	153.6	1/5	QPSK
DRC3	1024	4	3	204.8	1/5	QPSK
DRC3	1024	4	2	307.2	1/5	QPSK
DRC3	1024	4	1	614.4	8/21	QPSK
DRC4	1024	2	2	307.2	1/5	QPSK
DRC4	1024	2	1	614.4	8/23	QPSK
DRC5	1024	1	1	614.4	1/3	QPSK
DRC6	2048	4	4	307.2	1/3	QPSK
DRC6	2048	4	3	409.6	1/3	QPSK
DRC6	2048	4	2	614.4	1/3	QPSK
DRC6	2048	4	1	1228.8	16/23	QPSK
DRC7	2048	2	2	614.4	1/3	QPSK
DRC7/8	2048	1 or 2	1	1228.8	2/3	QPSK
DRC9	3072	2	2	921.6	1/3	8-PSK
DRC9/10	3072	1 or 2	1	1843.2	2/3	8-PSK
DRC11	4096	2	2	1228.8	1/3	16-QAM
DRC11/12	4096	1 or 2	1	2457.6	2/3	16-QAM

Note that in some of the higher bit rate cases there is some overlap in the formats. This is because the one-slot cases of DRC8, DRC10, and DRC12 are identical in terms of the transmitted packet as the first slot of DRC7, DRC9, and DRC11, respectively. Thus, the error rates of packets being sent by the former group of formats will be the same as that for the latter group of formats when attempting to decode the packet after one slot. The remaining lower bit rate formats are not equivalent, due to the varying length of preamble sent for each format.

After the received signal is de-spread, de-multiplexed, demodulated, and soft-combined, it is sent to a turbo decoder. This decoder is based on an iterative application of maximum *a posteriori* (MAP) decoding. An extensive analysis of the decoding process conducted by Ms. Ge Li of the University of

Alberta/TR*Labs* showed that 8 iterations of the turbo decoder were sufficient to reduce the packet error rates to an approximately steady-state value. That is, further iterations of the decoder would not significantly reduce the PERs beyond the values obtained after 8 iterations. The resulting PERs as a function of E_S/N_0 for the formats listed in Table III are shown in Appendix A.

4.3. Radio Environment Simulations

The model for the radio environment contains many details. Included are decisions on how the base stations and mobiles will be physically arranged, what type of environment the equipment operates in, and the effect of that environment on the resulting radio signals.

The physical geometry of the simulation is as recommended in [25], and shown in the figure below. There are a total of 19 cells, each with a base station transmitter located at the center of the cell. The 18 surrounding cells are arranged in two tiers around the central cell, with this last cell being of interest in the simulations. Each cell is divided into three hexagonal sectors, thereby creating a total of 57 sectors. Each sector has an antenna pointed along the radius of the sector as indicated by the arrows in the figure. Users are assumed to be distributed uniformly over the area of each of the sectors.

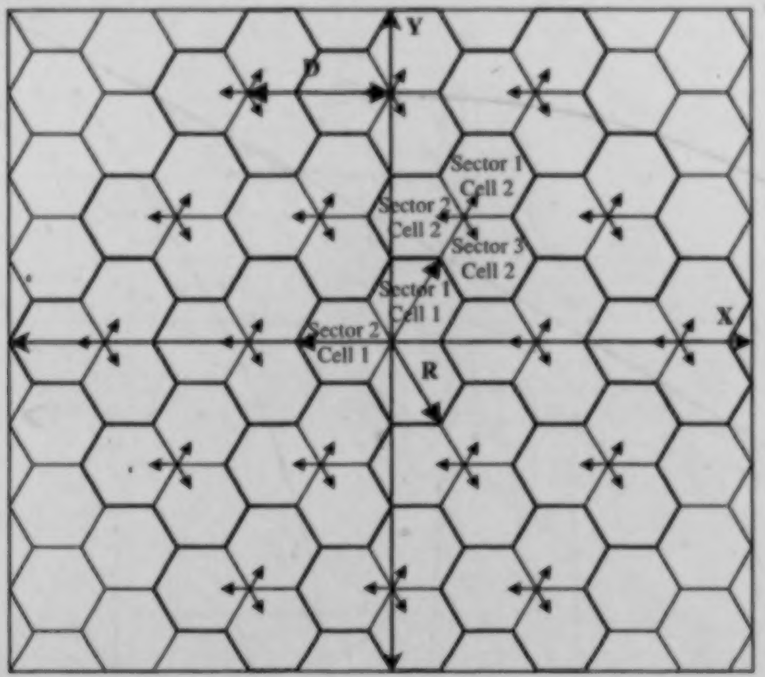


Figure 4.2: Cell layout used for simulations

The appropriate size for the cell radius (as denoted by R in the figure above) is determined using the COST231-Walfish-Ikegami path loss model [26]. This model is appropriate for urban environments and small cells with transmitter-receiver separations between 20 m and 5 km, and for carrier frequencies between 800 and 2000 MHz. For non line-of-sight (NLoS) propagation of signals (i.e. when buildings and other obstructions exist between the base station and the mobile), the total path loss L_p is given by the sum of three components: the free-space loss L_o, the roof-top-to-street diffraction and scatter loss L_{rts}, and the multi-screen diffraction loss L_{msd} (all in dB units). In a medium city and suburban area where the base station antennas are above the height of the building rooftops, the path loss can be calculated as follows:

$$L_p = L_o + L_{rts} + L_{msd} \qquad (4.3)$$

$$L_o = 32.4 + 20\log(d) + 20\log(f_c) \qquad (4.4)$$

$$L_{rts} = -16.9 - 10\log(w) + 10\log(f_c) + 20\log(h_{Roof} - h_m) + L_{ori} \qquad (4.5)$$

$$L_{msd} = 54 - 18\log(1 + h_b - h_{Roof}) + 18\log(d) + \left[-4 + 0.7\left(\frac{f_c}{925} - 1\right)\right]\log(f_c) - 9\log(b) \qquad (4.6)$$

The equations above are expressed as a function of the distance *d* in km, the carrier frequency f_c in MHz, the width of the streets *w* in m, the average height of the buildings h_{Roof} in m, the height of the mobile h_m in m, the height of the base station antennas h_b in m, and the building separation *b* in m. The term L_{ori} is an orientation loss based on the direction of the roads relative to the direct radio path, which is assumed to be zero. All other parameters are as suggested in [27] and given in the table below.

TABLE IV:
VALUES OF PARAMETERS USED FOR CELL RADIUS COMPUTATION

Parameter	Symbol	Possible Range	Value Used in Simulation
Distance from base station antenna	d	0.02 – 5 km	A variable; to be determined
Carrier frequency	f_c	800 – 2000 MHz	1960 MHz
Street width	w	–	20 m
Average height of buildings	h_{Roof}	4 – 50 m	18 m
Mobile height	h_m	1 – 3 m	1.5 m
Base station antenna height	h_b	–	21 m
Building separation	b	–	40 m
Orientation loss	L_{ori}	–	0 dB

Inserting the values into the above equations, and substituting R for *d*, we find that the path loss at the cell edge along the bore sight of the antenna is given by $L_p = 143.761 + 38\log(R)$.

It now remains to determine what the maximum allowable path loss can be. A base station can typically be expected to transmit with 17 Watts of power per sector. This value can also be seen in evaluation methods for other systems (e.g. [28], [29]). About 80% of that transmit power will be allocated to data transmissions, while the rest is allocated to other types of traffic. A minimum power of around –85.6 dBm is required for acquiring the signal at the receiver. (This includes a large-scale fading margin.) Furthermore, a commercially available antenna is used in the simulations; namely, specifications for the EMS Wireless model number RR65-18-00DP are used. This antenna has a nominal gain of 17.5 dBi and a 3-dB beamwidth of about 65 degrees. An antenna cable loss of 3 dB and a penetration loss of 10 dB are also assumed. The thermal noise power spectral density is assumed to be -174 dBm/Hz. However, for the bandwidth of 1.25 MHz of the 1xEV-DO system, this only contributes about 5 pW (–113 dBm) of noise power to the system, so thermal noise is negligible.

It should be checked if the received power along the bore sight of the antenna at the edge of the sector is indeed the minimum received power throughout the entire sector. Since a real antenna is assumed, its directivity pattern is available, which is shown in the figure below.

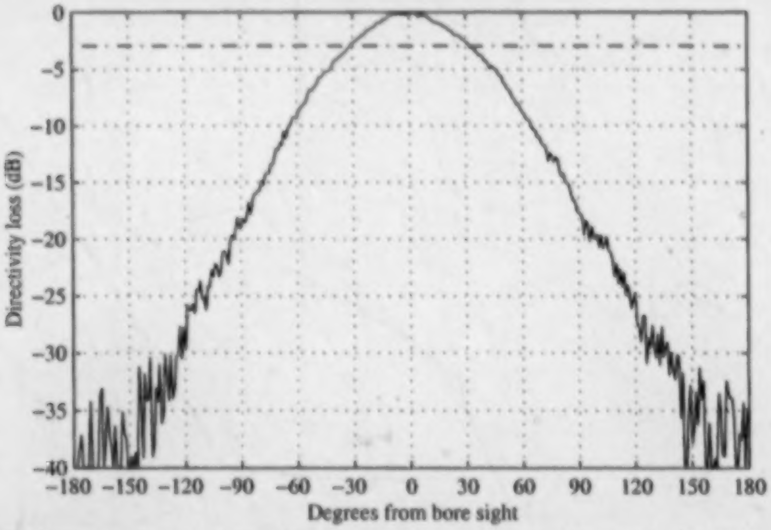


Figure 4.3: Directivity loss pattern versus angle from antenna bore sight for EMS Wireless RR65-18-00DP antenna

Taking this pattern into account, the actual received power can be calculated as a mobile receiver moves along the edge of a sector. Using a normalized cell radius of 1 km, the received power along the edge of a sector taking into account path loss and the antenna pattern is shown below. (Note that using a normalized radius of 1 km will not change the relative received power; a different value of the radius would simply scale the received power by a constant for all angles, so the relative power would remain the same.)

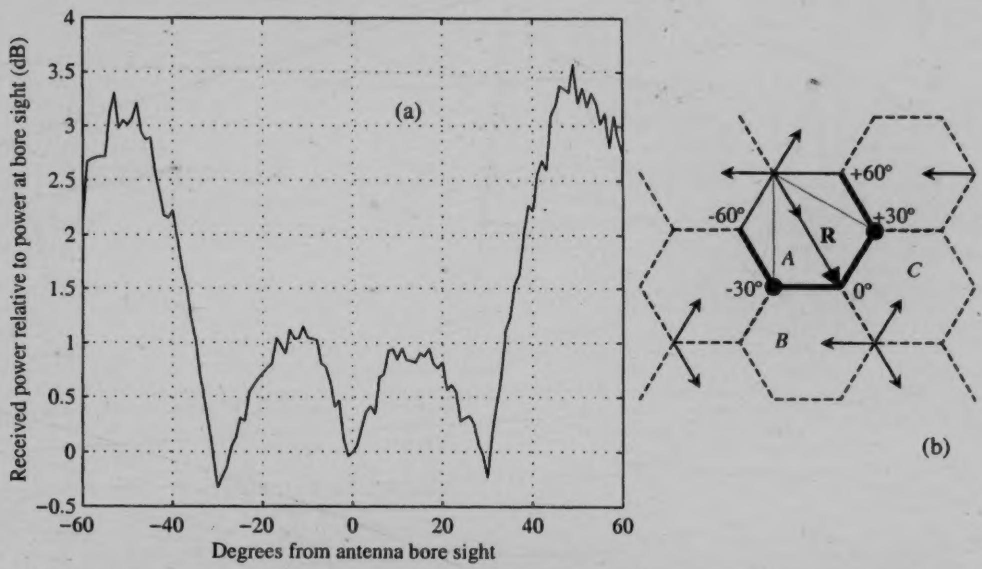


Figure 4.4: (a) Received power relative to the bore sight at 0° in dB versus the angle from the bore sight of the antenna if the mobile receiver is positioned at the border of a sector; (b) The perimeter of a sector, where the measurements for (a) would be taken

From part (a) of the figure above, it can be seen that once the antenna pattern is accounted for, the minimum power received along the sector perimeter is not in fact along the bore sight of the antenna, but rather at ±30° off the axis of the bore sight. The difference in the received power is about 0.3 dB, which should be taken into account when calculating the cell radius. Note that the difference is applicable for this particular path loss model and antenna. If the path loss or the antenna was changed, the minimum signal might end up being located along the bore sight.

Considering a multi-cell, multi-sector environment with best sector selection, not counting any fading effects, the weakest signal should still be expected at the ±30° points. As the mobile moves along the perimeter of the sector between those points, the mobile is physically closer to the transmitter in sector B (for –30° to 0°) or C (for 0° to 30 °) than it is to that in A. Even when the antenna pattern is considered, the signal strength between 0° and 30° would be the strongest from the transmitter in sector C than it would from sector A. Moving from 0° to 30°, the signal received from the transmitter in sector C would be equivalent to the signal received from sector A as the mobile moves from 60° to 30°. A similar argument can be made for sector B and the range from 0° to -30°. Thus, based on the path loss and antenna pattern alone, in a normal situation the mobile on the boundary of the sector would use transmitter B or C for reception rather than A, and so the middle part of the graph in part (a) of the figure above would never be seen. Rather, at those angles, it would be a mirror image of the outer half of the graph, as the received signal strength would take on those values due to the signals from sectors B and C.

Taking the various gains and losses into account, the maximum allowable path loss can be found to be $L_{p\_max} = 131.1$ dB. The details of this calculation can be found in the table below. Equating this value to the result from the Walfish-Ikegami path loss of $L_p = 143.761+38\log(R)$, the radius of the cell can be found to be R = 464 m. However, that value is probably a bit overly precise to be realistic. As such, a rounded value of 465 m shall be used for the simulations. This corresponds to a distance between base stations of D = 1.5R = 697.5 m.

TABLE V:
LINK BUDGET VALUES FOR THE FORWARD CHANNEL

Parameter	Value	Comments
(a) Transmission power per sector P_{t_max}	17 W = 42.3 dBm	–
(b) Effective data transmission power P_t	80% of P_{t_max} = 13.6 W = 41.3 dBm	–
(c) Base station antenna bore sight gain G_b	17.5 dBi	EMS Wireless RR65-18-00DP
(d) Antenna cable loss L_c	3 dB	–
(e) Total data transmission power along bore sight of antenna	(b)+(c)-(d)=55.8 dBm	–
(f) Penetration loss L_{pen}	10 dB	–
(g) Minimum sector power relative to bore sight	-0.3 dB	–
(h) Power spectral density of thermal noise	-174 dBm/Hz	Neglected
(i) Minimum required received power at mobile P_{rec_min}	-85.6 dBm	Includes large scale fading margin ($\sigma_X = 6.5$ dB)
(j) Maximum allowable path loss L_{p_max}	(e)-(f)+(g)-(i)= 131.1 dBm	–

Another large-scale effect seen on signals in a radio environment is shadow fading. This effect comes from the fact that users who are located at similar distances from the transmitter, but at different locations will experience different signal levels due to the variation in buildings and other obstructions between the transmitter and the receivers. Shadowing is modeled as an additive zero-mean lognormal Gaussian random process. The values generated by the Gaussian process are in units of dB and are added directly to the received signal strength after the path loss is considered. Thus, on a large scale (i.e. averaged over the course of several slots), the received signal power at a mobile from sector i is given by:

$$\overline{P}_{rec}^{i} = P_t + G_b + G_d^i(\theta) - L_{pen} - L_p^i(d) - X^i, \quad i = 1,2,\ldots,57 \tag{4.7}$$

In the above equation, P_t is the transmitted power, G_b is the antenna bore sight gain, $G_d^i(\theta)$ is the directivity loss of the antenna from sector i, based on the angle of the mobile from the bore sight of that antenna, L_{pen} is the penetration loss, L_p^i is the COST231 Walfish-Ikegami path loss from the transmitter in sector i (at a distance of d), and X^i is the shadowing variable for sector i. The shadowing variable has an associated standard deviation σ_X. In a typical micro-cellular environment, the standard deviation can range from 4 to 13 dB [26]. A value of $\sigma_X = 6.5$ dB was used for the simulations.

The shadow fading process experiences spatial auto-correlation. That is, the values of the shadow fading terms are correlated in different locations in space. This correlation must be taken into account when the mobile is in motion. In [30], Gudmundson proposed a discrete-time exponential model for the spatial auto-correlation of shadow fading:

$$\Phi(k) = E\left[X_d^i X_{d+k\Delta d}^i\right] = \sigma_X^2 \exp\left(-\frac{|k|\Delta d \ln 2}{d_{dec}}\right) = \sigma_X^2 \exp\left(-\frac{|k| v\Delta t \ln 2}{d_{dec}}\right) \quad (4.8)$$

In the above equation, k is the discrete-time step number, Δd is the distance moved during every step, Δt is the time taken between updates of the shadowing, v is the mobile velocity, and d_{dec} is the decorrelation distance (i.e. the distance for which the auto-correlation drops to 50%). In [31] and [25], values of $d_{dec} = 20$ m and 5 m are suggested for vehicular and pedestrian environments, respectively. However, since the simulations will be dealing with two pedestrian channels with differing time-dispersion properties, it was decided to use $d_{dec} = 5$ m for the less dispersive "A" channel and $d_{dec} = 10$ m for the more dispersive "B" channel. It was also decided to update the shadowing variable at intervals of $\Delta d = d_{dec}/10$ (i.e. every 360, 720, and 72 slots for the pedestrian A, pedestrian B, and vehicular channels, respectively). These channels will be discussed in more detail later.

The exponential auto-correlation function can be realized using a first-order discrete-time recursive filter [26]:

$$X_{k+1}^i = \xi X_k^i + (1-\xi)\upsilon_k^i \quad (4.9)$$

Here, X_{k+1}^i is the updated shadow term for base station i at position k+1 that is correlated with the previous value X_k^i at position k, ξ is a parameter used to control the spatial auto-correlation, and υ_k is a discrete-time, zero-mean white Gaussian process with standard deviation σ_υ. It can be shown that the auto-correlation function of the process at the output of the filter is:

$$\Phi(n) = E\left[X_k^i X_{k+n}^i\right] = \frac{1-\xi}{1+\xi}\sigma_\upsilon^2 \xi^{|n|} \quad (4.10)$$

Comparing the auto-correlation of the filter with the desired equation obtained by Gudmundson, the two will be equivalent if the following substitutions are made:

$$\xi = \exp\left(-\frac{\Delta d \ln 2}{d_{dec}}\right); \quad \sigma_\upsilon^2 = \frac{1+\xi}{1-\xi}\sigma_X^2 \quad (4.11)$$

In addition to the spatial auto-correlation, there is a cross-correlation between the lognormal shadowing variables of two different base stations j and k. This fact is of interest for the simulations, as it is primarily the other cells that are the source of interference for the sector of interest. Therefore, the cross-correlation must be accounted for in order to properly calculate the inter-cell interference. This cross-correlation is denoted by α, and is defined as follows:

$$\alpha \equiv \frac{E\left[X^j X^k\right]}{\sigma_{X^j}\sigma_{X^k}}, \quad j=1,2,...,19, \quad k=1,2,...,19, \quad j \neq k \tag{4.12}$$

In the simulations, a value of $\alpha = 0.5$ is assumed for the cross-correlation of shadowing between different base stations. For different sectors of the same base station, the cross-correlation is 1 [26]. In order to create the proper cross-correlation between the random variables X^i, each variable is generated as a linear combination of two independent and identically distributed Gaussian variables Z and Z^i (such that $E[ZZ^i] = 0$ and $E[Z^iZ^j]=0$ for $i \neq j$), each with zero mean and a standard deviation of $\sigma_Z = \sigma_X$. The variable Z is common to all base stations.

$$X^i = aZ + bZ^i, \quad i=1,2,..,19 \tag{4.13}$$

By taking $a = b = 1/\sqrt{2}$, the generated variables X^i will have the desired cross-correlation and standard deviation.

To generate the desired cross- and auto-correlated sequences of shadow fading values, first a sequence of 19 cross-correlated υ_k^i values with standard deviation σ_υ are created:

$$\upsilon_k^i = \frac{1}{\sqrt{2}}Z + \frac{1}{\sqrt{2}}Z^i, \quad i=1,2,...,19; \quad \sigma_Z = \sigma_\upsilon \tag{4.14}$$

These 19 values are then used to create the desired auto-correlated shadowing variable sequences:

$$X_{k+1}^i = \xi X_k^i + (1-\xi)\upsilon_k^i, \quad i=1,2,...,19 \tag{4.15}$$

The following figure shows a comparison of the exponential auto-correlation function and the auto-correlation of 760,000 shadow fading values created by simulation using the method described above. A standard deviation of $\sigma_X = 6.5$ dB and $\Delta d = d_{dec}/10$ are assumed.

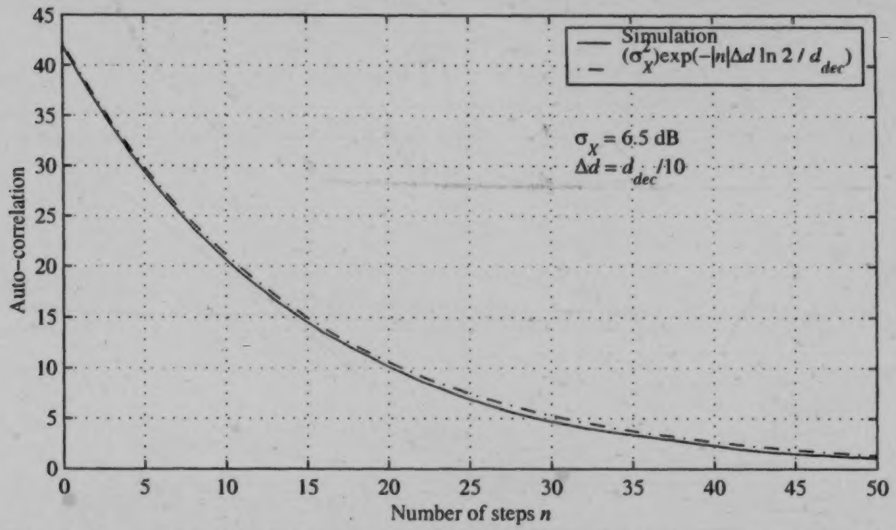


Figure 4.5: Comparison of simulated and postulated exponential auto-correlation of 760,000 lognormal shadow fading values

The motion of the mobile receiver and movement of the scatterers result in a multipath environment. Due to the obstacles in the environment between the transmitter and the mobile receiver, the signal can be reflected and scattered. Consequently, multiple copies of the transmitted signal can arrive at the receiver at different times. The phases from each of the paths will be random, and so the combination of the signals will cause rapid variations in the total received signal strength. Hence, fading on a small scale (i.e. from slot to slot) is seen. This small-scale fading is modeled as a Rayleigh random process. The Rayleigh variable can be viewed as the variation in the envelope of a complex Gaussian random process. If the complex Gaussian variable can be given by $x_R + jx_I$, where x_R and x_I are zero-mean independent and identically distributed (i.i.d.) Gaussian variables with variance σ^2, then the Rayleigh variable would be $r = \sqrt{x_R^2 + x_I^2}$. The probability density function of the envelope is [32]:

$$p_r(r) = \frac{r}{\sigma^2}\exp\left(-\frac{r^2}{2\sigma^2}\right), \quad r \geq 0 \tag{4.16}$$

Alternately, the variation in the power of the Rayleigh process (i.e. the square of the envelope) follows an exponential distribution:

$$p_\gamma(\gamma) = \frac{1}{\overline{\gamma}}\exp\left(-\frac{\gamma}{\overline{\gamma}}\right), \quad \gamma \geq 0 \tag{4.17}$$

In the above equation, $\overline{\gamma} = 2\sigma^2 = P_{rec}$ is the average received envelope power. In the simulation, an exponentially distributed random variable with mean envelope power P_{rec} is obtained by multiplying P_{rec} by a normalized exponential random variable γ with a mean of $\overline{\gamma} = 1$. The exponentially distributed random variable is obtained by squaring the Rayleigh distributed random variable. On a dB scale, this means that the received power from a sector is given by:

$$P_{rec}^{i} = P_t + G_b + G_d^i(\theta) - L_{pen} - L_p^i(d) - X^i + \gamma_{(\mathrm{dB})}, \quad i = 1,2,\ldots,57 \qquad (4.18)$$

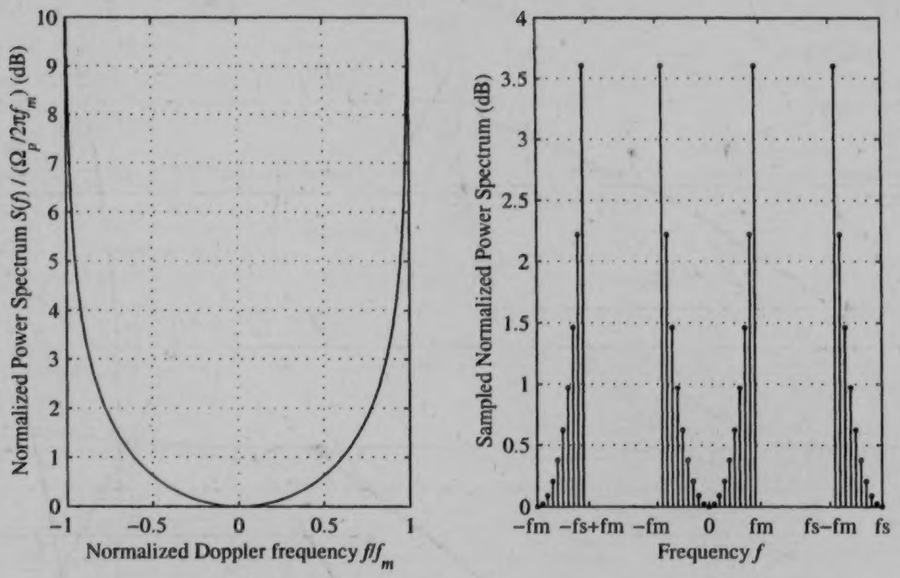
The Rayleigh fading process exhibits an auto-correlation and spectral shape that can be described using Clarke's model of isotropic scattering ([24],[26],[33]). This model assumes that the field incident on the mobile antenna is composed of a number of azimuthal plane waves with arbitrary carrier phases and azimuthal angles of arrival. Each wave is assumed to have an equal average amplitude, as in the absence of a line-of-sight path, the scattered components will experience similar attenuations due to the small differences in distances travelled. In this model, the auto-correlation of the real and imaginary parts $x_R(t)$ and $x_I(t)$ of the complex envelope is given by [26]:

$$\phi_{x_R x_R}(\tau) = \phi_{x_I x_I}(\tau) = \frac{\Omega_p}{2} J_0(2\pi f_m \tau) \qquad (4.19)$$

Here, Ω_p is the total received envelope power, $J_0(\cdot)$ is the zero-order Bessel function of the first kind, and $f_m = vf_c/c$ is the maximum Doppler frequency for the mobile. Note that for the received bandpass signal $r(t)$, $E[r^2(t)] = \Omega_p/2$, and f_m equals 5.44 Hz and 108.89 Hz at speeds of $v = 3$ km/h and $v = 60$ km/h respectively at a carrier frequency of $f_c = 1960$ MHz. The power spectral density of $x_R(t)$ and $x_I(t)$ is given by [26]:

$$S_{x_R x_R}(f) = S_{x_I x_I}(f) = \begin{cases} \dfrac{\Omega_p}{2\pi f_m} \dfrac{1}{\sqrt{1-(f/f_m)^2}} & |f| \le f_m \\ 0 & \text{otherwise} \end{cases} \qquad (4.20)$$

This spectrum is shown in the figure below. For reference, a sampled version of the spectrum (sampled at frequency f_s) is also shown.



(a) (b)

Figure 4.6: (a) Normalized power spectral density of Rayleigh fading process; (b) Sampled version of normalized power spectral density. f_s = sampling frequency; f_m = Doppler frequency

In order to generate a Rayleigh random process with the above autocorrelation and spectrum, a modified Smith method is used, as described by Young and Beaulieu in [34]. In this method, a complex Gaussian white noise process is sampled and filtered to obtain the desired spectrum. The filtered samples then undergo an inverse discrete fast Fourier transform (IDFFT) to obtain the correlated complex Gaussian noise sequence in the time domain. The magnitude of the time-domain samples gives the Rayleigh process. A block diagram for the modified Smith method is shown below. The modified method is more computationally efficient than the standard Smith method, which has two branches similar to the entire diagram shown below.

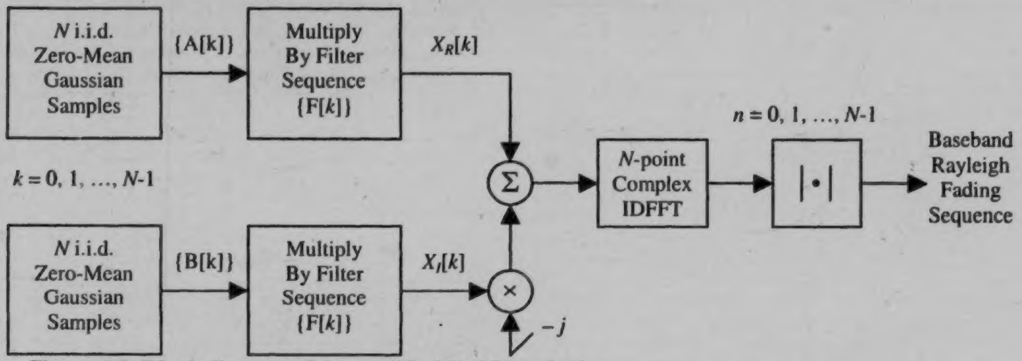

N i.i.d. Zero-Mean Gaussian Samples
{A[k]}
Multiply By Filter Sequence {F[k]}
$X_R[k]$
$k = 0, 1, \ldots, N$-1
Σ
N-point Complex IDFFT
$n = 0, 1, \ldots, N$-1
$|\cdot|$
Baseband Rayleigh Fading Sequence
N i.i.d. Zero-Mean Gaussian Samples
{B[k]}
Multiply By Filter Sequence {F[k]}
$X_I[k]$
×
$-j$

Figure 4.7: Block diagram of the modified Smith method

Since the power spectral density of a discrete-time process is periodic with a frequency equal to the sampling frequency f_s (f_s = 1/1.666… ms = 600 Hz in the simulations) instead of sampling the Gaussian process in the range of $|f| \leq f_m$, one can instead sample in the ranges $0 \le f \le f_m$ and $(f_s - f_m) \le f \le f_s$. These samples are then multiplied by the coefficients of a low-pass filter as follows:

$$\{F[k]\} = \begin{cases} 0, & k = 0 \\ \sqrt{\dfrac{1}{2\sqrt{1-\left(\dfrac{k}{f_k N}\right)^2}}}, & k = 1, 2, \ldots, k_m - 1 \\ \sqrt{\dfrac{k_m}{2}\left[\dfrac{\pi}{2} - \arctan\left(\dfrac{k_m - 1}{\sqrt{2k_m - 1}}\right)\right]}, & k = k_m \\ 0, & k = k_m + 1, \ldots, N - k_m - 1 \\ \sqrt{\dfrac{k_m}{2}\left[\dfrac{\pi}{2} - \arctan\left(\dfrac{k_m - 1}{\sqrt{2k_m - 1}}\right)\right]}, & k = N - k_m \\ \sqrt{\dfrac{1}{2\sqrt{1-\left(\dfrac{N-k}{f_k N}\right)^2}}}, & k = N - k_m + 1, \ldots, N - 2, N - 1 \end{cases} \qquad (4.21)$$

For the above, $f_k = f_m/f_s$ is the ratio of the maximum Doppler frequency to the sampling frequency, and $k_m = \lfloor f_k N \rfloor$, where N is the number of sampling points in the frequency domain. A value of N of $2^{17} = 131{,}072$ was used for a mobile speed of 3 km/h, and $N = 2^{15} = 32768$ was used for 60 km/h.

At the output of the IDFFT, the real and imaginary parts are given by:

$$x_c(n) = x_R(n) + jx_I(n) \qquad (4.22)$$

The term n refers to steps of the sampling period $t_s = 1/f_s$. The auto-correlation of the real and imaginary parts is given by:

$$\phi_{x_R x_R}(n) = \phi_{x_I x_I}(n) = J_0(2\pi f_m n t_s) \qquad (4.23)$$

The above method was used to create 9 million samples of the Rayleigh process. The statistics were normalized so as to have $2\sigma^2 = \overline{\gamma} = \Omega_p = 1$. The following diagram shows a comparison of the probability density functions of the simulated values from the modified Smith method and their squares (for the distribution of the power of the process), and the theoretical Rayleigh and exponential distributions.

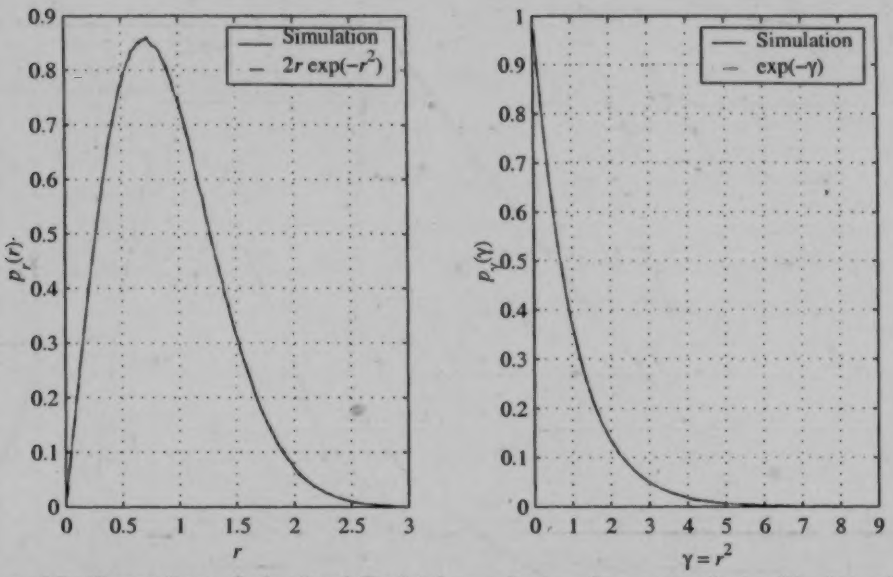


Figure 4.8: Comparison of simulated Rayleigh envelope and power values with theoretical probability density functions

As can be seen from the figure above, there is a very close match between the distributions of the theoretical and simulated data; the curves overlap each other. To further check the distributions, ¼ of the 9 million Rayleigh samples and their squares (the smaller data set was due to memory limitations) were passed to a MATLAB function called "weibfit". This function takes a set of values and, using the assumption that the data follows a Weibull distribution, does a maximum likelihood estimation of the Weibull distribution parameters *a* and *b*. The function also provides the 95% confidence intervals of the parameters. The Weibull distribution is given by:

$$p_x(x) = abx^{b-1}\exp\left(-ax^b\right) \qquad (4.24)$$

It can be seen that the Rayleigh distribution with $\sigma^2 = ½$ is a special case of the Weibull distribution when $a = 1$ and $b = 2$, while the corresponding exponential distribution is a special case when $a = b = 1$. Using the simulated Rayleigh values, the "weibfit" function gave values of $a = 1.011$ and $b = 2.004$, with intervals of 1.009 to 1.012 and 2.002 to 2.007, respectively. The squared values provided parameters of $a = 1.011$ and $b = 1.002$, with intervals of 1.009 to 1.012 and 1.001 to 1.003, respectively. Therefore, there is a strong match between the theoretical distributions and the distributions of the simulated samples.

The following two figures show a comparison between the normalized auto-correlation of the real part of the output of the IDFFT block $x_R(n)$ in the simulation and the ideal autocorrelation function. 9 million values were used for both the 3 km/h and the 60 km/h cases. There is once again an extremely close match between the simulations and the theoretical formula, so much so that the two lines on both graphs essentially overlap. It should also be observed that the auto-correlation function decreases much faster in time at the higher mobile velocity, which is a direct consequence of the auto-correlation formula as given earlier.

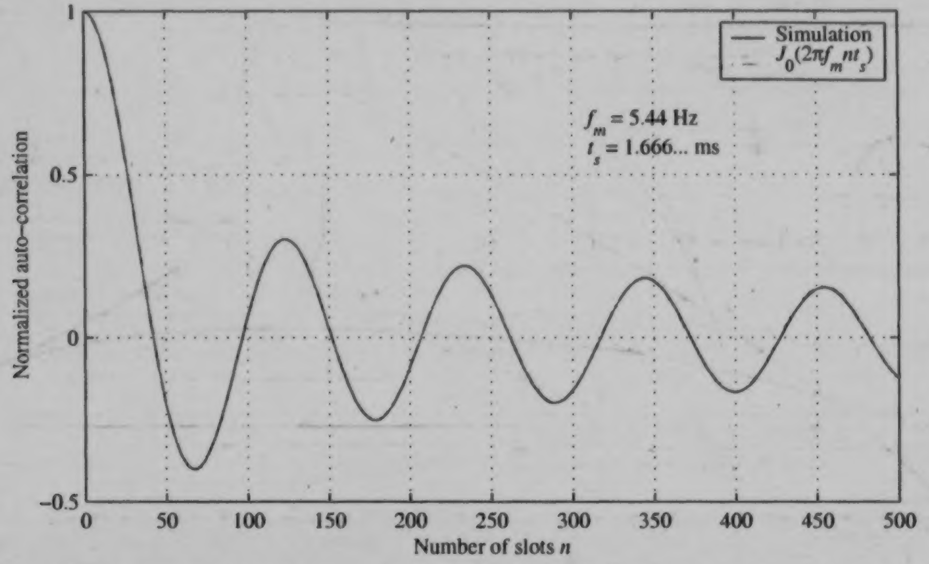


Figure 4.9: Simulated and ideal auto-correlation of Rayleigh fading process at a mobile speed of 3 km/h and carrier frequency of 1960 MHz

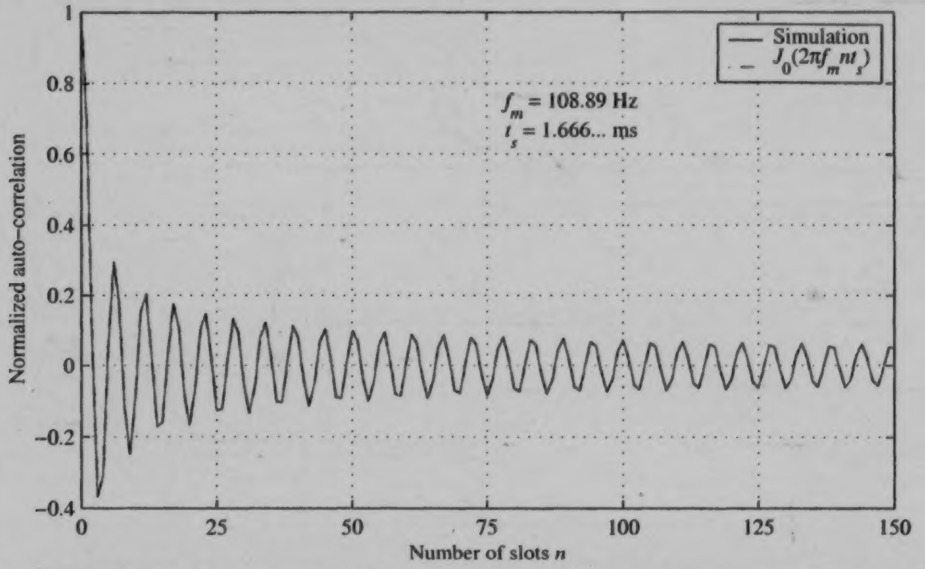


Figure 4.10: Simulated and ideal auto-correlation of Rayleigh fading process at a mobile speed of 60 km/h and carrier frequency of 1960 MHz

Another effect of multipath propagation is frequency selectivity, or channel dispersion. Frequency-selective fading is when the bandwidth of the transmitted signal is greater than the channel coherence bandwidth (that is, the frequency range over which the channel appears approximately constant). It occurs when the difference between the arrival times of the paths in a multipath environment is at least a significant fraction of the duration of the signaling element in a system. For CDMA systems, the signaling elements are the chips used to spread the signal. Thus, if the time differential of the paths is greater than the chip period, frequency-selective fading will be seen. If the differential is smaller, flat fading will be seen.

In CDMA systems, a RAKE receiver can help to mitigate the effects of dispersion. Thanks to the spreading code used in CDMA systems, the RAKE receiver can resolve the delayed versions of the signal and combine the paths together to improve the received signal. The individual delayed paths can be resolved and captured by the fingers of the RAKE receiver so long as the delay between the paths is greater than the chip period. The 1xEV-DO system operates at a chip rate of 1.2288 Mcps, so the chip period is approximately $T_c = 814$ ns. We assume in this simulation that the RAKE receiver has three fingers (i.e. it can capture up to three resolvable paths).

To model the multipath environment, the definitions for the ITU pedestrian A, ITU pedestrian B, and ITU vehicular B channels were used as given in [31]. The three channels are characterized by a respectively increasing delay spread in the paths of the channels. The pedestrian channels are assumed to have an associated mobile speed of 3 km/h, while the vehicular channel assumes a mobile speed of 60 km/h, as mentioned earlier. The delay and power profiles of the channels are given in the table below.

TABLE VI:
DELAY AND POWER PROFILES OF THE ITU PEDESTRIAN A, ITU PEDESTRIAN B, AND ITU VEHICULAR B CHANNELS

Path	ITU pedestrian A		ITU pedestrian B		ITU vehicular B	
	Relative Delay (ns)	Average Power (dB)	Relative Delay (ns)	Average Power (dB)	Relative Delay (ns)	Average Power (dB)
1	0	0	0	0	0	-2.5
2	110	-9.7	200	-0.9	300	0
3	190	-19.2	800	-4.9	8900	-12.8
4	410	-22.8	1200	-8.0	[illegible]	[illegible]
5	–	–	[illegible]	[illegible]	17100	-25.2
6	–	–	3700	-23.9	20000	-16.0

The power gains of the received paths are normalized such that the total power received by all the paths is equal to unity.

When considering the power captured by the RAKE receiver, it is assumed that the fingers of the receiver can achieve a perfect synchronous lock with a path. It is also assumed that in the event that paths are not resolvable, the finger will lock onto the strongest path in the set of non-resolvable paths. The remaining paths in the set will have their power reduced based on the auto-correlation $R_x(\tau)$ of the spreading sequence. The auto-correlation is approximately triangular, and can be expressed by:

$$R_x(\tau) \equiv \begin{cases} 1-\dfrac{|\tau|}{T_c}, & -T_c \le \tau \le T_c \\ 0, & \text{otherwise} \end{cases} \qquad (4.25)$$

Here, τ is the time offset of the weaker path from the strongest path in the set of non-resolvable paths.

For the ITU pedestrian A channel, there are a total of 4 paths; however, none of the paths are resolvable within the chip period of 814 ns. As a result, they effectively become a single non-resolvable path. The RAKE receiver locks onto the strongest path at 0 ns, with the second path being reduced according to its time offset. The other two paths are neglected in the captured power due to their small relative power values. Since the time offset of the second path is 110 ns, the reduction of the power is:

$$R_x(110) = 1 - \frac{110}{814} = 0.864832 \qquad (4.26)$$

Therefore, the fraction of the total received power that is captured by the RAKE receiver is:

$$\frac{\text{Captured power}}{\text{Total received power}} = \frac{10^{0/10} + 0.864832 \cdot 10^{-9.7/10}}{10^{0/10} + 10^{-9.7/10} + 10^{-19.2/10} + 10^{-22.8/10}} \approx 0.97 \qquad (4.27)$$

Therefore, in the ITU pedestrian A channel, about 97% of the total received power is captured by the RAKE receiver.

In the ITU pedestrian B channel, 6 paths are specified. Paths 1 and 2 form one non-resolvable set of paths, paths 3 and 4 form a second set of non-resolvable paths, and path 5 acts as a single resolvable path. Path 6 is neglected in the captured power both due to its low relative power and the assumption that the RAKE receiver has only 3 fingers. The time offset of 200 ns of path 2 from the stronger path 1 means that its power is reduced by a factor of 0.75424 as it is captured. Similarly, the power on path 4 is reduced by a factor of 0.50848 due to the 400 ns offset from path 3. Therefore, the fraction of the total received power in each of the three fingers as follows:

$$\frac{P_{f1}}{\mathrm{P}_{rec}} = \frac{10^{0/10} + 0.75424 \cdot 10^{-0.9/10}}{10^{0/10} + 10^{-0.9/10} + 10^{-4.9/10} + 10^{-8.0/10} + 10^{-7.8/10} + 10^{-23.9/10}} = 0.6544 \qquad (4.28)$$

$$\frac{P_{f2}}{\mathrm{P}_{rec}} = \frac{10^{-4.9/10} + 0.50848 \cdot 10^{-8.0/10}}{10^{0/10} + 10^{-0.9/10} + 10^{-4.9/10} + 10^{-8.0/10} + 10^{-7.8/10} + 10^{-23.9/10}} = 0.164 \qquad (4.29)$$

$$\frac{P_{f3}}{\mathrm{P}_{rec}} = \frac{10^{-7.8/10}}{10^{0/10} + 10^{-0.9/10} + 10^{-4.9/10} + 10^{-8.0/10} + 10^{-7.8/10} + 10^{-23.9/10}} = 0.067 \qquad (4.30)$$

Thus, in the ITU pedestrian B channel, about 65.44% of the received power is captured by finger 1, about 16.4% is captured by finger 2, and about 6.7% is captured by finger 3.

In the ITU vehicular B channel, paths 1 and 2 form a set of non-resolvable paths, and paths 3 and 4 act as individual resolvable paths. The remaining paths are not captured. In the non-resolvable set of paths, the first finger of the RAKE is assumed to lock onto path 2 at 300 ns. Consequently, path 1 has its power reduced by a factor of 0.63136. Overall, the fractional powers captured by the three fingers are:

$$\frac{P_{f1}}{\mathrm{P}_{rec}}=\frac{0.63136\cdot 10^{-2.5/10}+10^{0/10}}{10^{-2.5/10}+10^{0/10}+10^{-12.8/10}+10^{-10.0/10}+10^{-25.2/10}+10^{-16.0/10}}=0.7774 \quad (4.31)$$

$$\frac{P_{f2}}{\mathrm{P}_{rec}}=\frac{10^{-12.8/10}}{10^{-2.5/10}+10^{0/10}+10^{-12.8/10}+10^{-10.0/10}+10^{-25.2/10}+10^{-16.0/10}}=0.030 \quad (4.32)$$

$$\frac{P_{f3}}{\mathrm{P}_{rec}}=\frac{10^{-10.0/10}}{10^{-2.5/10}+10^{0/10}+10^{-12.8/10}+10^{-10.0/10}+10^{-25.2/10}+10^{-16.0/10}}=0.057 \quad (4.33)$$

Thus, in the ITU vehicular B channel, about 77.74% of the received power is captured by finger 1, about 3% is captured by finger 2, and about 5.7% is captured by finger 3.

To determine the final value of the SIR for each mobile in every slot, first the serving sector must be determined. This is the sector that the mobile would request data from, and is determined at the beginning of every slot based on the sector with the largest total received power, based on the path loss and shadowing components of the signal alone.

In the ITU pedestrian A channel, the captured power from the serving sector *i* can be expressed on a linear scale by:

$$P^{i}_{cap\_k}=0.97\bar{P}^{i}_{rec\_k}\left(R^{i}_{k}\right)^{2}=0.97P^{i}_{rec\_k} \quad (4.34)$$

In other words, the captured power is simply 97% of the total received power. The total received power consists of two components: a Rayleigh distributed variable R^{i}_{k} to account for small-scale fading (squared so as to consider power variations instead of envelope variations), and the power $\bar{P}^{i}_{rec\_k}$ due to path loss and shadowing. As given earlier, $\bar{P}^{i}_{rec\_k}$ can be found on a decibel scale by:

$$\bar{P}^{i}_{rec\_k(\mathrm{dB})}=P_{t(\mathrm{dB})}+G_{b(\mathrm{dB})}+G^{i}_{d}\left(\theta_{k}\right)_{(\mathrm{dB})}-L_{pen(\mathrm{dB})}-L^{i}_{p}\left(d_{k}\right)_{(\mathrm{dB})}-X^{i}_{k(\mathrm{dB})} \quad (4.35)$$

The interference power seen in every slot is simply the sum of the total received powers from the non-serving sectors:

$$I_{k}=\sum_{\substack{j=1\\ j\neq i}}^{57}P^{j}_{rec\_k}=\sum_{\substack{j=1\\ j\neq 1}}^{57}\bar{P}^{j}_{rec\_k}\left(R^{j}_{k}\right)^{2} \quad (4.36)$$

The signal-to-interference ratio SIR_k is then simply the ratio of $P^{i}_{cap\_k}$ to I_k, or 10 times the base-10 logarithm of the ratio for the SIR in dB.

In the ITU pedestrian B channel, the situation is somewhat more complicated. Each of the three fingers in the RAKE receiver of mobile *k* captures part of the received power. The power in each of the three fingers is:

$$P^{1i}_{cap\_k} = 0.6544\overline{P}^{i}_{rec\_k}\left(R^{1i}_k\right)^2 \tag{4.37}$$

$$P^{2i}_{cap\_k} = 0.164\overline{P}^{i}_{rec\_k}\left(R^{2i}_k\right)^2 \tag{4.38}$$

$$P^{3i}_{cap\_k} = 0.067\overline{P}^{i}_{rec\_k}\left(R^{3i}_k\right)^2 \tag{4.39}$$

Each of the three paths in the channel experiences independent small-scale fading, and so each of the three paths is multiplied by an independent Rayleigh fading contribution.

Considering the interference power, in addition to the combined powers from the non-serving sectors, the power not captured by each of the fingers interferes with the captured signals. Note that the time offset of non-resolvable paths does not cause a reduction in the interference power like it does in the captured power.

Consider the first finger in the pedestrian B channel. Because of the time offset, the power of the second path is reduced by a factor of 0.75424. However, the entire power of that path causes interference on the other fingers. Therefore, the interference power of the paths captured at finger 1 on the other fingers is greater than the captured power of finger 1 by a factor of:

$$\frac{\text{Interference power}}{\text{Captured power}} = \frac{10^{0/10} + 10^{-0.9/10}}{10^{0/10} + 0.75424 \cdot 10^{-0.9/10}} = 1.12 \tag{4.40}$$

In a similar manner, it can be found that the interference power of the paths captured at finger 2 on fingers 1 and 3 is 1.19 times the captured power of finger 2. Since finger 3 only consists of one path, the captured power of the path at finger 3 is identical to the interference power of that path on fingers 1 and 2. The total interference powers at the three fingers can be expressed by:

$$I^1_k = I_k + 1.19P^{2i}_{cap\_k} + P^{3i}_{cap\_k} \tag{4.41}$$

$$I^2_k = I_k + 1.12P^{1i}_{cap\_k} + P^{3i}_{cap\_k} \tag{4.42}$$

$$I^3_k = I_k + 1.12P^{1i}_{cap\_k} + 1.19P^{2i}_{cap\_k} \tag{4.43}$$

The term I_k is the same as in the pedestrian A channel.

It is assumed that the RAKE receiver combines the signals received in the three fingers using maximal ratio combining. In this method, the signals of the three fingers are weighted proportionally to the signal-to-interference ratio received at each finger. The result is that the overall SIR is simply the sum of the

SIRs on the individual fingers. In dB, the final SIR at the output of the receiver is given by:

$$SIR_k = 10\log\left(\frac{P^{1i}_{cap\_k}}{I^1_k} + \frac{P^{2i}_{cap\_k}}{I^2_k} + \frac{P^{3i}_{cap\_k}}{I^3_k}\right) \qquad (4.44)$$

The SIR values for the vehicular B channel are computed in much the same way as for the pedestrian B channel. The main difference is the fraction of the total power captured by each of the fingers.

$$P^{1i}_{cap\_k} = 0.7774\overline{P}^{i}_{rec\_k}\left(R^{1i}_k\right)^2 \qquad (4.45)$$

$$P^{2i}_{cap\_k} = 0.030\overline{P}^{i}_{rec\_k}\left(R^{2i}_k\right)^2 \qquad (4.46)$$

$$P^{3i}_{cap\_k} = 0.057\overline{P}^{i}_{rec\_k}\left(R^{3i}_k\right)^2 \qquad (4.47)$$

It can also be shown that the interference power of the paths captured at finger 1 on fingers 2 and 3 is 1.15 times the captured power in finger 1. The interference powers at the fingers can therefore be expressed as:

$$I^1_k = I_k + 1P^{2i}_{cap\_k} + P^{3i}_{cap\_k} \qquad (4.48)$$

$$I^2_k = I_k + 1.15P^{1i}_{cap\_k} + P^{3i}_{cap\_k} \qquad (4.49)$$

$$I^3_k = I_k + 1.15P^{1i}_{cap\_k} + P^{2i}_{cap\_k} \qquad (4.50)$$

The RAKE receiver uses maximal ratio combining to obtain the final SIR values the same as in the pedestrian B channel.

In the vehicular B channel, there is very little correlation in the Rayleigh fading variable after a few slots. Therefore, it makes little sense to attempt to use that Rayleigh component when attempting to predict future SIR values. Consequently, the vehicular B simulations also generate SIR values to be used for prediction or format determination purposes. These SIR_p values do not include the Rayleigh component of the signals in the calculations. The values are essentially calculated the same way as for the regular SIR values, except that the R_k variables are removed from the equations (or equivalently, the R_k terms are set to a constant of 1). The issue of prediction and format determination will be discussed in more detail in the section on system level simulations.

The following figure shows a cumulative distribution function (CDF) of the final SIR and SIR_p values obtained from the radio environment simulations. A total of 7.2 million SIR values were used to obtain each of the CDFs.

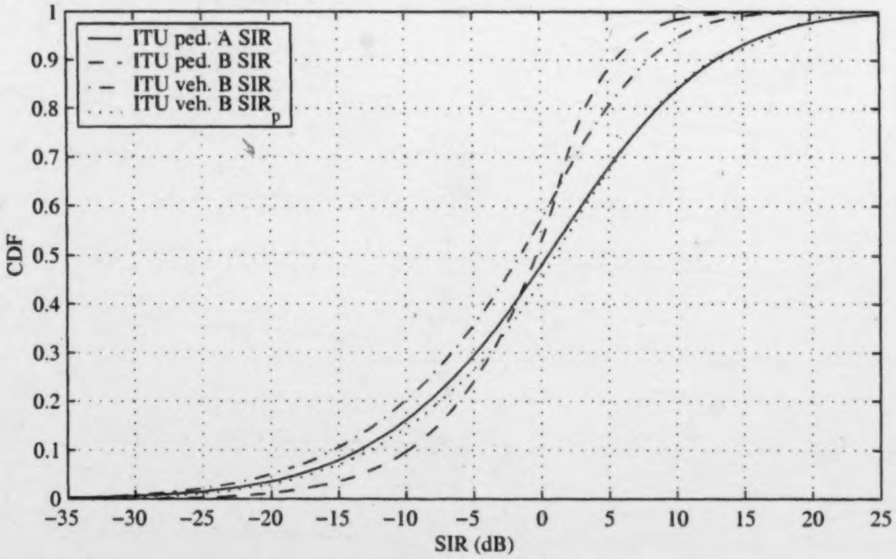


Figure 4.11: CDFs of SIR values obtained from radio environment simulations

4.4. System Level Simulations

The system level simulations are used to generate the final throughput, delay, and fairness statistics for each of the scheduling algorithms being tested. To begin, users are dropped uniformly over the area of a sector in the central cell in groups of 1, 2, 4, 8, 16, and 32 users per drop. 100 drops are performed, with 30 seconds (18000 slots) of simulations per drop. It is assumed that all the users have infinite packet queues, such that packets are always available to be sent to each user. Furthermore, it is assumed that all the users request delay-tolerant data, such that there are no Quality of Service (QoS) specifications that must be met, and that no one user has a higher priority to receive data service than any other user.

The radio environment simulations are used to create SIR values for each user in every slot for every drop. The values, which are stored in a lookup table, are then read into the system level simulations. As the values are read in, they are modified in order to account for imperfections seen in non-ideal systems. Some of these imperfections include inter-chip interference, a loss of code channel orthogonality due to transmitter non-linearities, non-ideal waveform quality, analog-to-digital conversion quantization noise, adjacent channel interference, and so on. The effect of these imperfections has been measured by various research groups, and can be expressed in terms of a signal power to imperfection-interference power ratio SIR_I. Measurement results reported by Ericsson Wireless [35] and Qualcomm [36] are given in the following table. By normalizing the signal power to 1, the imperfection-interference power can be obtained for each effect. The individual values can then be summed to find the overall total effect SIR_I^T.

TABLE VII:
THE SIR_I RESULTING FROM SYSTEM IMPERFECTIONS

Imperfection	SIR_I
Inter-chip interference	16.5
Loss of channel orthogonality	29.0
Non-ideal waveform quality	18.1
ADC quantization noise	20.0
Adjacent channel interference	27.0
Total: SIR_I^T	12.9

Because there are likely some additional imperfection effects that were not accounted for, the value for SIR_I^T will be rounded up to 13 dB. This rounded value also agrees with a recommendation in [37], which states that expecting a maximum C/I of 13 dB is reasonable. Using this value, the values for SIR and SIR_p determined in the previous section are modified according to:

$$SIR_M = \frac{SIR}{1+\frac{SIR}{\alpha}} \qquad (4.51)$$

In the above equation, $\alpha = 10^{SIR_I^T/10} = 10^{1.3}$. This modification has the effect of saturating the maximum SIR received at 13 dB. Large values of SIR are reduced to approximately the maximum value, while small SIR values are changed very little. The following figure shows the CDFs of the SIR and SIR_p values after being modified.

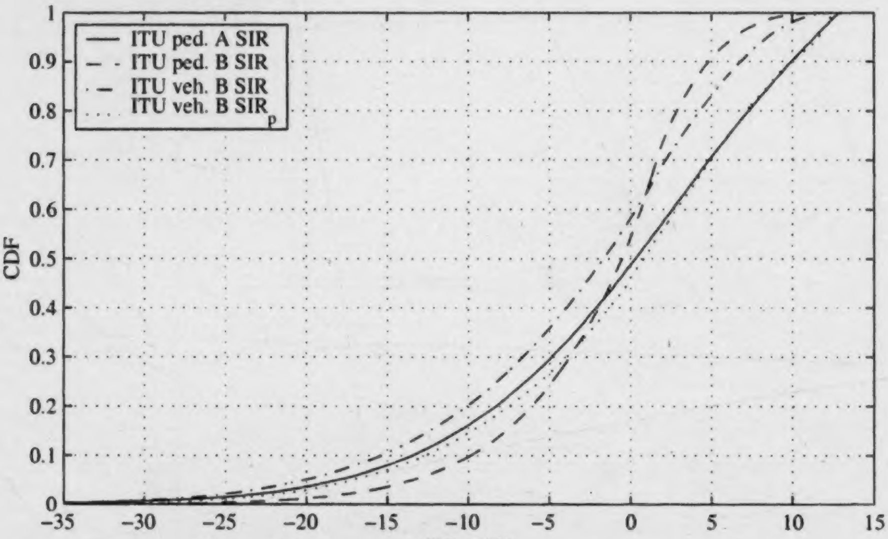


Figure 4.12: CDFs of modified SIR values after accounting for system imperfections

Once the SIR values are read in to the simulations, the formats that the mobiles can support must be determined. In the 1xEV-DO system, the time between the pilot bits being received, the SIR being estimated, the format being selected, the DRC request being sent to the base station, and the base station selecting a user for transmission is approximately 3 slots [10]. Therefore, it would be desirable to predict what the value of the SIR will be 3 slots in the future. However, the concept of channel prediction is a complex field in and of itself, and is beyond the scope of this work. An overview of this topic can be found in [38].

Rather than deal with the issues surrounding prediction, two simplified cases are considered. First, rather than attempt any prediction, instead the estimate of the future SIR value is the current modified SIR value (or SIR_p value in the ITU vehicular B channel) with a margin subtracted:

$$\widehat{SIR}(n+3) = SIR_M(n) - \Delta_{1\%} \qquad (4.52)$$

The margin $\Delta_{1\%}$ is tuned so as to provide approximately a 1% packet error rate (PER) for all formats being transmitted (i.e. about 1% of the packets transmitted using each format will be in error) and a median PER per user of about 1%. The process of tuning the margin so as to obtain the target PER is discussed in the performance evaluation chapter.

The other SIR estimation method is based on perfect prediction. In theory, a predictor can determine what will happen to the transmitted signal up to a wavelength in the future. This is equivalent to a period of time that is the inverse of the Doppler frequency. At a carrier frequency of 1960 MHz, this is equal to 184 ms (about 110 slots) for the pedestrian channels at 3 km/h, and 9 ms (about 5.5 slots) for the vehicular channel at 60 km/h. Note that the farthest ahead a mobile would have to predict would be 63 slots. This is under the assumption that a packet is sent using the lowest possible rate and takes the maximum number of 16 re-transmissions to be received. Since the re-transmissions occur 4 slots apart, the last re-transmission would occur 63 slots after the time that the initial prediction is made. Therefore, it is theoretically possible for the mobile to predict the SIR value during all possible transmission slots in a pedestrian channel. In the vehicular channel, it is only feasible to predict what the SIR value will be for the first transmission slot 3 slots in the future. Nevertheless, it is also of interest to see what would happen if all possible transmission slots could be predicted. Thus, for the perfect prediction case, we have two scenarios. For the pedestrian channels, the predicted SIR value will be the average SIR over the possible future transmission slots, based on perfect knowledge of the pertinent SIR_M values. For the vehicular channel, both the average SIR over all possible future transmission slots and the SIR for the first transmission slot will be used, again using perfect knowledge of the future channel conditions including the imperfections.

Note that the actual modified SIR values are used for perfect prediction, and not the SIR_p values in the vehicular channel. Also note that for perfect one-slot prediction, a margin must still be subtracted to obtain the target 1% PER. In the case of perfect multislot prediction, the number of slots that the SIR is averaged over depends on the format being tested. When checking if DRC*x* can be supported, the value used is:

$$\widehat{SIR}_{(\mathrm{dB})} = 10\log\left(\frac{1}{N_{\max}}\sum_{k=3}^{4N_{\max}-1} SIR(k)\right), \quad k = 3,7,11,15,\ldots \qquad (4.53)$$

The values of $SIR(k)$ are on a linear scale instead of a decibel scale. The value $N_{\max}$ is the maximum possible number of slots that can be used to send a packet using format DRCx.

When determining what format can be supported, the system level simulations use as a reference the required E_S/N_0 for 1% PER in the AWGN channel, assuming the maximum number of transmission slots are used. These values can be found from the AWGN PER curves, and are given in the table below.

TABLE VIII:
REQUIRED E_S/N_0 TO OBTAIN 1 % PER IN AN AWGN CHANNEL

Format	Nominal Data Rate (kb/s)	Maximum number of slots	Minimum required E_S/N_0 (dB) for 1% PER in AWGN channel
DRC1	38.4	16	-13.5
DRC2	76.8	8	-10.5
DRC3	153.6	4	-7.4
DRC4	307.2	2	-4.3
DRC5	614.4	1	-1.0
DRC6	307.2	4	-4.2
DRC7	614.4	2	-1.2
DRC8	1228.8	1	3.7
DRC9	921.6	2	1.5
DRC10	1843.2	1	7.1
DRC11	1228.8	2	3.4
DRC12	2457.6	1	9.2

The estimated $\widehat{SIR}$ is compared with the values of $E_S/N_{0(1\%)}$, and the format with the highest data rate that satisfies $\widehat{SIR} \geq E_S/N_{0(1\%)}$ is selected as the format that the mobile requested from the base station. Note that the formats sharing the same data rates also have similar values of $E_S/N_{0(1\%)}$. For these cases, only the formats with the smaller number of transmission slots (i.e. DRC4, DRC5, and DRC 8) are allowed in the pedestrian channels, and only the formats with the larger number of transmission slots (i.e. DRC6, DRC7, and DRC11) are allowed in the vehicular channel. The larger number of slots provides more diversity in the rapidly fading vehicular channel, and hence gives a better possibility that the packet will be received correctly. The larger number of slots is not necessary in the slower fading pedestrian channels.

After a mobile is selected for transmission by the scheduling algorithm during a given slot, it must be determined if the transmission was successful or not. It is difficult to determine what the actual performance of the turbo decoder will be when the re-transmissions, which arrive at different SIR levels, are combined. To simplify the matter, the SIR_M value is averaged on a linear scale over the number of slots the packet transmission has taken thus far, and that average is assumed to be equivalent to the E_S/N_0 had the transmission taken place in an AWGN channel. (Note that the SIR_M values are used in all cases, including the vehicular channel. The modified SIR_p values are only used during the format determination process in the "no prediction" case.) The averaged SIR value is then compared with the AWGN PER curves for the format and number of transmission slots in question to obtain the corresponding PER. Linear interpolation is used between the actual simulated E_S/N_0 points on the curves. This process provides the probability of the mobile returning a NAK for the packet. To determine if the packet is actually decoded correctly or not, a uniformly distributed random number between 0 and 1 is generated. If the number is less than the PER, the packet is not decoded correctly, and a NAK is returned; otherwise, the packet is decoded correctly, and an ACK is returned. If the averaged SIR is below the lower range of the PER curve, the packet is always NAKed. Likewise, if the averaged SIR is above the upper range of the PER curve, the packet is always ACKed.

As a final step, the system level simulations output the raw data concerning the user and packet data rates, the user and packet delays, the number of allocated slots per user, and the number of correctly/incorrectly received packets per format and per user at the end of each drop. This data is analyzed in the performance evaluation chapter.

Chapter 5: Performance Evaluation

5.1. Introduction

The performance evaluation takes the raw data outputted from the system level simulations and obtains useful information regarding the performance of the scheduling algorithm being tested. The following performance measures are presented in this work:

- Average sector throughput as a function of the number of users per sector: This measure takes the average throughput for each user over the course of a 30-second (i.e. 18000-slot) drop and sums them to obtain the total sector throughput for that drop. The values from the 100 drops are then averaged to obtain the average sector throughput. The average sector throughput is plotted versus the number the users in the sector for all the scheduling algorithms being analyzed.
- Distributions of average throughput per user: For the case of 16 users per sector, the average throughput per user for each of the 16 users and each of the 100 drops is obtained (for a total of 1600 values). The spread of the distributions of the throughputs is an indication of how fairly the scheduling algorithm treats the various users. The 16-user case is used as it provides the best indication of the actual distribution of values due to the relatively large number of values. At the same time, it ensures that the values are significant as each user receives a reasonable number of packets. As the number of users increases beyond 16 per sector, there is a concern that all users may not receive a sufficient number of packets for the distributions to be accurate. Using the 16-user case avoids this problem.
- Distributions of effective bit rates: The actual bit rate achieved by a packet depends on the format used and the number of re-transmissions to receive the packet correctly. This measure looks at the distribution of effective bit rates achieved across all packets transmitted by a given scheduling algorithm.
- Distributions of formats and number of slots used: Similar to the distributions of effective bit rates, this measure looks at the proportion of packets transmitted using each format for a given scheduling algorithm. Within each format, the number of slots used to receive packets under that format is also shown.
- Average delay per packet per drop as a function of the number of users per sector: For a given drop, the delays of the packets regardless of which user received them are averaged. That average value is then averaged over all the drops. The final value is plotted versus the number the users in the sector for all the scheduling algorithms being analyzed. The delay for a packet is defined as the time between that packet arriving at the head of the queue for a user until that packet is either acknowledged, or the packet is discarded because it has used the maximum number of re-transmissions and a negative acknowledgement is still received for the packet. Note that by this definition, the minimum possible delay for a packet is 4 slots. Assuming that the

scheduling algorithm schedules two packets to the same user in back-to-back slots, the second packet will have a head-of-line queue delay of 1 slot, plus a further delay of 3 slots as the base station waits for the acknowledgement, for a total of 4 slots of delay.

- Average packet delay per user as a function of the number of users per sector: This measure first averages the packet delays seen by a given user during a given drop. The mean of those averages is then found over all users and drops, and plotted versus the number the users per sector for all the scheduling algorithms being analyzed. In general, these values can be significantly different than the average delay per packet, due to the differing number of packets received by each user. Consider an extremely lopsided 2-user case, in which one user receives a large number of packets, while the other receives only a few packets. The delays seen by the packets of the first user will be fairly small, while the delays seen by the packets of the second user will be quite large. Over the set of all packets, there is a large number of packets with a small delay, and a small number of packets with a large delay. Therefore, the average delay over all the packets will still be quite small. However, if a two-step average is done first over each given user, then over the set of users, the average delay seen per user will be much higher. In general, the average delay per packet asks, "If I am a packet being transmitted in a system with a given scheduling algorithm and number of users per sector, what kind of delay can I expect before I am received?" In comparison, the average delay per user asks, "If I am a user being served by a given scheduling algorithm, and there are a given number of other users also being served, what kind of delay can I expect that my packets will experience?"
- Distribution of delays per transmitted packet: For 16 users per sector, given the delays experienced by all packets for all users over all drops, this measure shows the distribution of those delays. Only packets that are transmitted during the 30-second drops are considered; if a user is starved for service during the entire drop, or if a given packet is not transmitted by the end of a drop, then that packet filtered out, as it is impossible to tell what the actual total delay of those packets would eventually be.
- Distribution of average delays per user: For the case of 16 users per sector, the average packet delay is found for each of the 16 users and the 100 drops. The distributions of the resulting 1600 values are then plotted. Only users who receive at least one packet are considered. This gives the probability of a given average delay, assuming that a user receives data in the first place.
- Distribution of allocated slots per user per drop: Once again for the 16-user case, the distribution of the number of slots allocated to each user per drop across all drops is plotted. This measure is an indication of how fairly the scheduling algorithm shares the total system usage time across all users requesting service.

Before any analysis can be performed, useful data must first be acquired. The process of tuning the margins for each transmission format in order to obtain these results is described in the next section.

5.2. Process of Deciding to Use Individual, Iterative Margins per Transmission Format

Before actually using the results from the system level simulations, it is first necessary to make sure that the results that are obtained are actually useful. In the case of these simulations, this means tuning the margins used for format determination in order to obtain the desired 1% PER.

To begin, a constant margin of 1 dB was assumed. Using this margin, the simulations were run for the round robin (RR), maximum DRC (maxD), and proportionally fair (PF) algorithms in the ITU pedestrian A channel. The resulting average PER per user is shown below as a function of the number of users per sector.

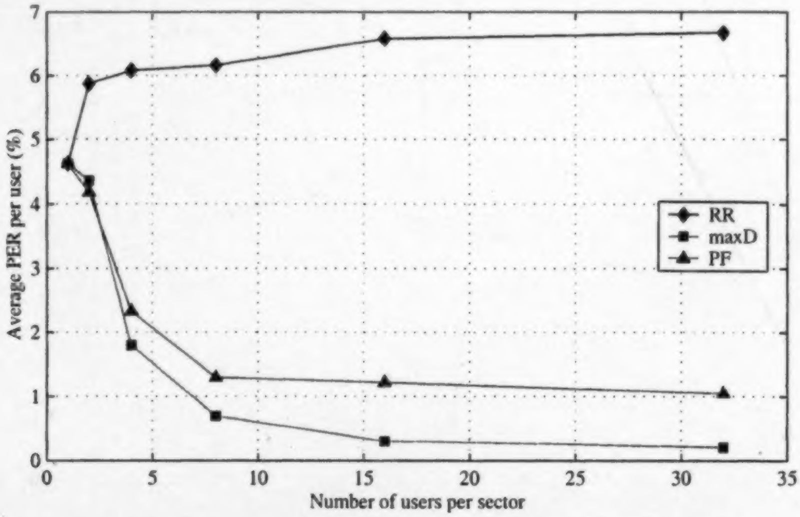


Figure 5.1: Average PER per user vs. number of users per sector for ITU pedestrian A channel using 1 dB margin for format determination

As can be seen by the graph, the RR algorithm yields the highest average PER per user, with values well above 1%. Therefore, while adjusting the margin, only the RR algorithm will be considered. Clearly, since the PERs are too large, the margin must be revised upwards. The following figure shows the average PER per user as a function of the number of users per sector as the margin is increased towards 3 dB.

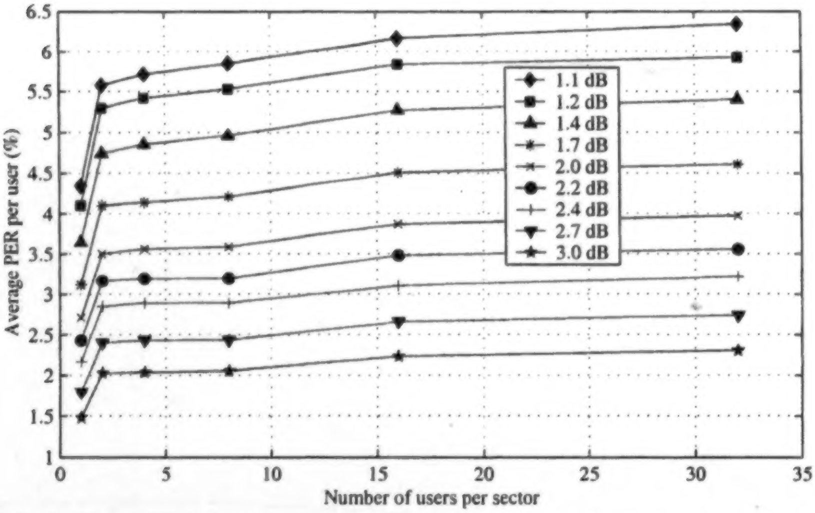


Figure 5.2: Average PER per user vs. number of users per sector for ITU pedestrian A channel using 1.1 dB to 3.0 dB margin for format determination

Even with a margin of 3 dB, the average PER per user is still too high. Therefore, a margin of 4 dB is attempted next. This margin was also used for the maxD and PF algorithms. The effect on the average PER per user for the three algorithms is shown below.

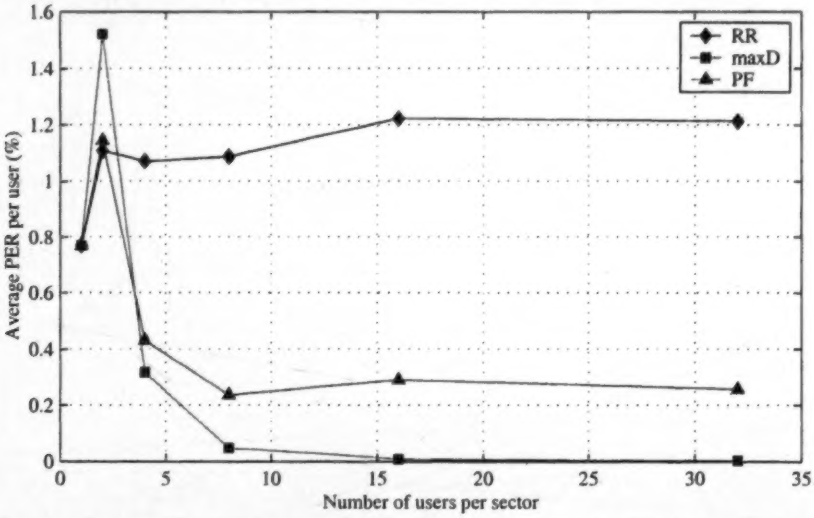


Figure 5.3: Average PER per user vs. number of users per sector for ITU pedestrian A channel using 4.0 dB margin for format determination

With a margin of 4 dB, the RR algorithm now is around the range of the desired 1% PER, although it still is a little high. However, in comparison, the maxD and PF algorithms show packet error rates much lower than the target 1%. From these margin changes, three observations can be made:

1. The margin needed to obtain the target packet error rate would seem to depend on the scheduling algorithm being used.
2. The margin also appears to depend on the number of users in the sector. Note for example, that to obtain an average PER per user of about 1.2%, a margin of 4 dB is needed for the PF algorithm with 2 users per sector, but only 1 dB is needed for 16 users per sector. Similarly, with the RR algorithm, a margin of 2.2 dB gives an average PER of 3.5% with 16 users per sector, but for 2 users per sector, 2.0 dB is required.
3. The margin of 4 dB is almost but not quite large enough to obtain the desired 1% PER. However, at this margin, the fastest data rate has been cut off. The highest possible SIR that a mobile will experience is 13 dB. The threshold of (SIR – margin) for the mobile to request DRC12 is 9.2 dB. However, because of the 4 dB margin, the largest value of (SIR – margin) is 9 dB, meaning that DRC12 will never be chosen, even though it likely can be supported.

Because of these points, especially the last one, a single margin does not appear to be the best solution to obtain the desired packet error rates. Rather, a margin that depends on the format being tested may be a more appropriate method. That is, when checking if DRC1 can be supported, the mobile would examine (SIR – Δ_{DRC1}) and compare that difference to the AWGN DRC1 1% PER threshold (i.e. –13.5 dB). When testing DRC2, the mobile would compare (SIR – Δ_{DRC2}) to –10.5 dB, and so on. This also has an advantage in that if a certain format is to be disallowed (e.g. DRC6, DRC7, and DRC11 in the pedestrian channels), it is only necessary to set the margin for that format to a very large value, such as 200 dB. In that case, the difference of SIR – 200 dB would never be above the threshold for that format, and so the given format would never be requested. (Note that in all cases here, the SIR value includes the 13 dB effect of imperfections as described in the previous chapter. This would be SIR_M as given earlier, but to save on notation, the M subscript has been dropped.)

To find an appropriate value for the margin, as a first step, the CDFs for the average SIR over all potential transmission slots less the current SIR, given that the current SIR can support a given format will be examined. That is, for DRC1, the CDF of $\{\text{AVG}[\text{SIR}(n+3), \text{SIR}(n+7), \ldots, \text{SIR}(n+63)] - \text{SIR}(n) \mid \text{SIR}(n) > -13.5 \text{ dB}\}$ will be examined; for DRC2, the CDF is of $\{\text{AVG}[\text{SIR}(n+3), \text{SIR}(n+7), \ldots, \text{SIR}(n+31)] - \text{SIR}(n) \mid \text{SIR}(n) > -10.5 \text{ dB}\}$; for DRC3, the CDF is of $\{\text{AVG}[\text{SIR}(n+3), \text{SIR}(n+7), \text{SIR}(n+11), \text{SIR}(n+15)] - \text{SIR}(n) \mid \text{SIR}(n) > \text{SIR}_{1\%}\}$, where $\text{SIR}_{1\%}$ is –7.4 dB; for DRC4 and DRC9, the CDF is of $\{\text{AVG}[\text{SIR}(n+3), \text{SIR}(n+7)] - \text{SIR}(n) \mid \text{SIR}(n) > \text{SIR}_{1\%}\}$, where $\text{SIR}_{1\%}$ is –4.3 dB and 1.5 dB, respectively; for DRC5, DRC8, DRC10, and DRC12, the CDF is of $\{\text{SIR}(n+3) - \text{SIR}(n) \mid \text{SIR}(n) > \text{SIR}_{1\%}\}$, where $\text{SIR}_{1\%}$ is –1.0 dB, 3.7 dB, 7.1 dB, and 9.2 dB, respectively. Note that the averaging of the SIR values is done on a linear scale, and then the average is converted back to dB. Also note that DRC6, DRC7, and DRC11 are not examined, since the ITU pedestrian A channel is being considered. The CDFs are shown in the figure below.

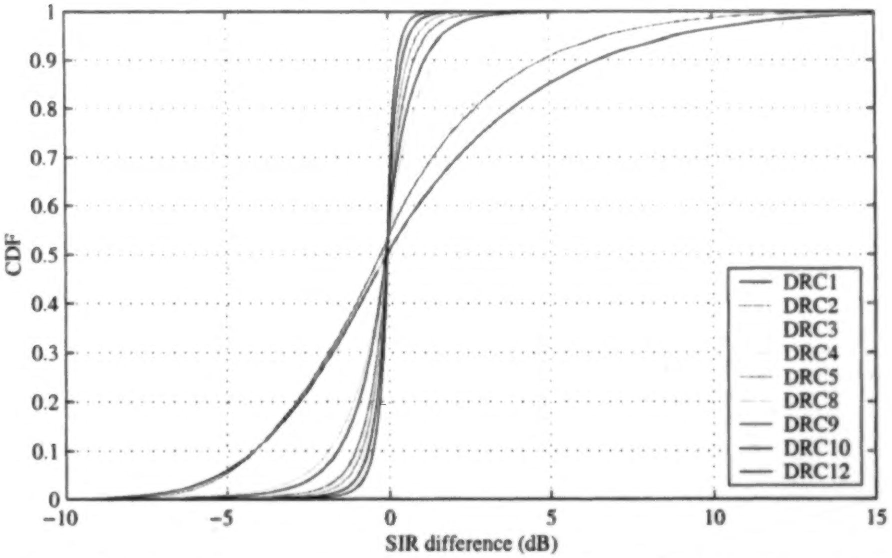


Figure 5.4: CDFs for all formats in ITU pedestrian A channel of differences between averaged SIR level over all possible transmission slots and current SIR level, conditioned on current SIR level being above that required for 1% PER.

Now, consider the 1% level of the CDFs. This point, always at an SIR difference below zero, indicates the 1% probability of the average SIR over the slots where a transmission would potentially take place being a certain number of dB below the current value of SIR. Stated another way, given the current level of SIR, the 1% point on the CDFs indicates the value *m* where there is a 1% probability of the average SIR being greater than *m* dB below the current level of SIR. The value *m* would be a good initial value for the margin for each DRC. For example, for DRC1, the 1% SIR difference is -7.25 dB. Therefore, a good initial value for the margin for DRC1 is *m*=7.25 dB.

This process must be iterated several times for each DRC. When generating a new CDF, the function, instead of being conditioned on $\{\mathrm{SIR}(n) > \mathrm{SIR}_{1\%}\}$, becomes conditioned on $\{\mathrm{SIR}(n) > (\mathrm{SIR}_{1\%} + m)\}$. Each iteration then provides a new value for *m*, until a steady-state value for *m* is reached (i.e. the value of *m* does not significantly change). The following graphs show this iteration process for the various DRCs used in the ITU pedestrian A channel.

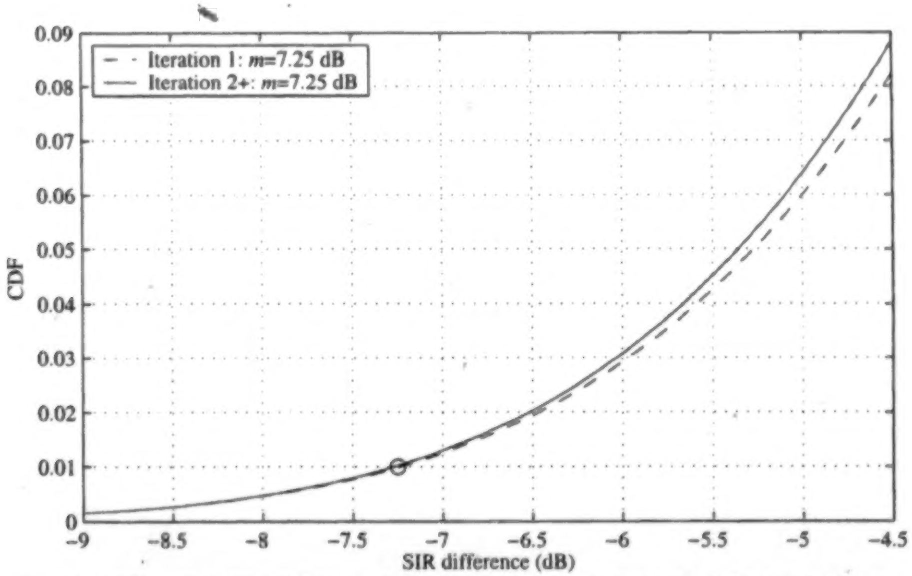


Figure 5.5: CDFs of *F*{AVG[SIR(*slot*+3),SIR(*slot*+7),…,SIR(*slot*+63)]–SIR(*slot*) | SIR(*slot*) > (-13.5+*m*) dB} with iterations of *m* based on 1% CDF level for DRC1 in ITU pedestrian A channel

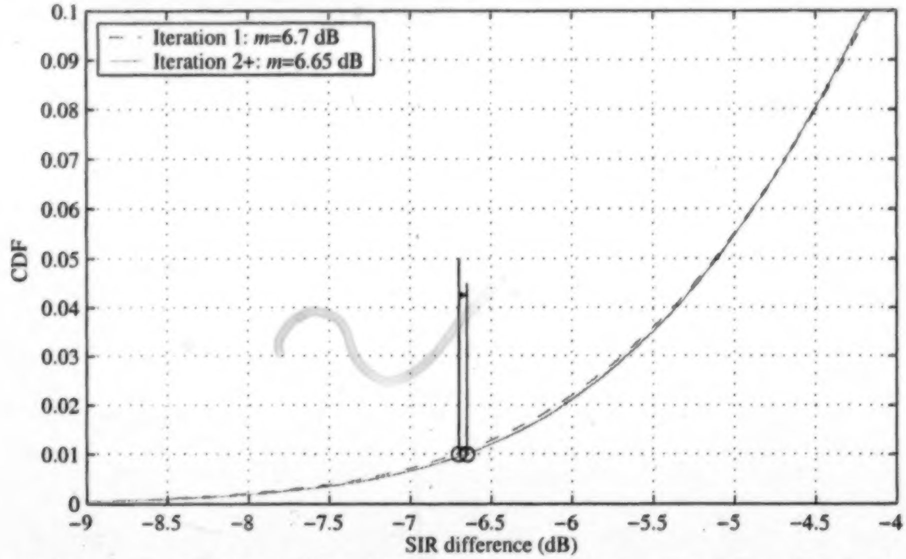


Figure 5.6: CDFs of *F*{AVG[SIR(*slot*+3),SIR(*slot*+7),…,SIR(*slot*+31)]–SIR(*slot*) | SIR(*slot*) > (-10.5+*m*) dB} with iterations of *m* based on 1% CDF level for DRC2 in ITU pedestrian A channel

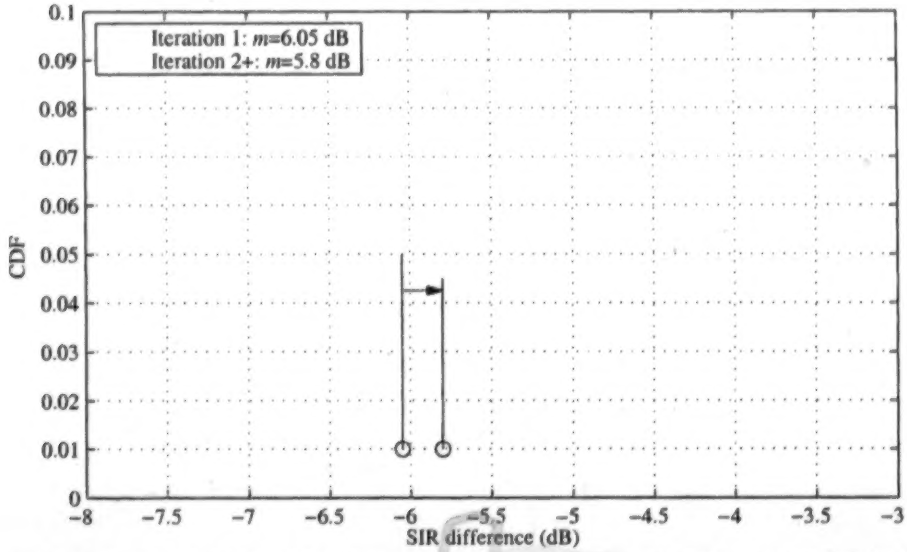


Figure 5.7: CDFs of $F\{$AVG[SIR(*slot*+3),SIR(*slot*+7), SIR(*slot*+11),SIR(*slot*+15)]–SIR(*slot*) | SIR(*slot*) > (-7.4+*m*) dB$\}$ with iterations of *m* based on 1% CDF level for DRC3 in ITU pedestrian A channel

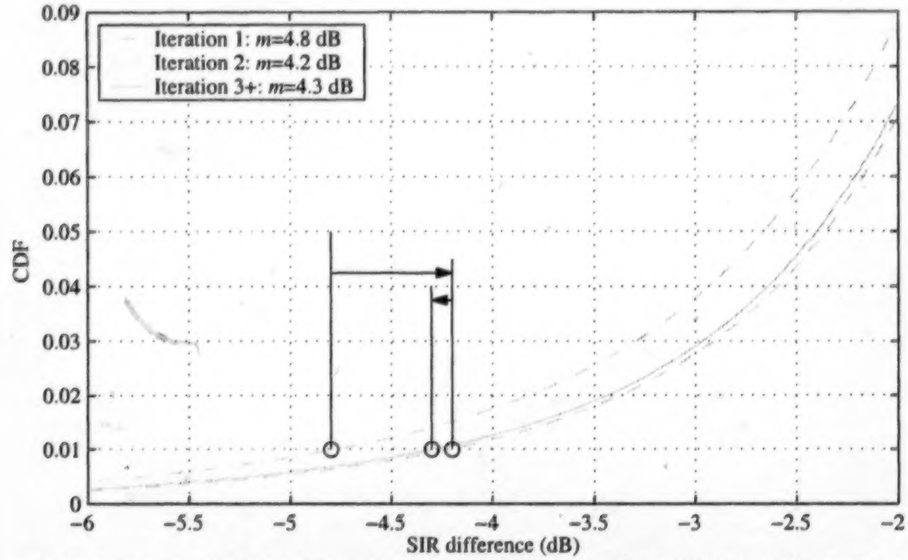


Figure 5.8: CDFs of $F\{$AVG[SIR(*slot*+3),SIR(*slot*+7)]–SIR(*slot*) | SIR(*slot*) > (-4.3+*m*) dB$\}$ with iterations of *m* based on 1% CDF level for DRC4 in ITU pedestrian A channel

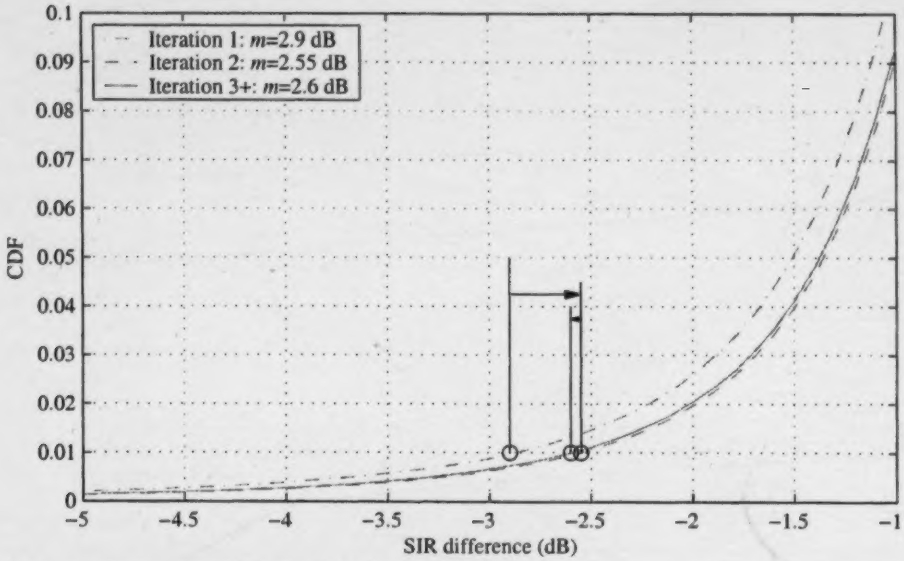


Figure 5.9: CDFs of $F\{\mathrm{SIR}(slot+3)–\mathrm{SIR}(slot) \mid \mathrm{SIR}(slot) > (-1.0+m)\ \mathrm{dB}\}$ with iterations of m based on 1% CDF level for DRC5 in ITU pedestrian A channel

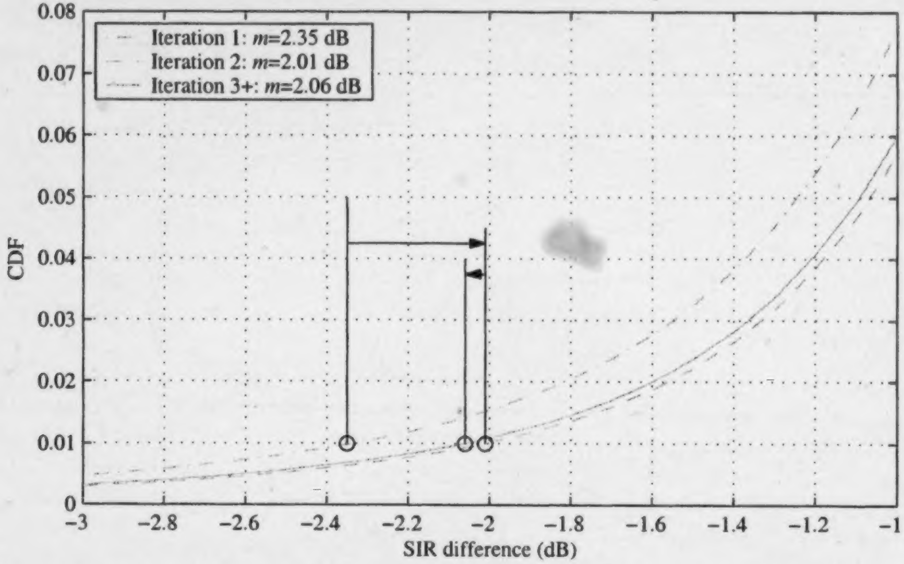


Figure 5.10: CDFs of $F\{\mathrm{SIR}(slot+3)–\mathrm{SIR}(slot) \mid \mathrm{SIR}(slot) > (3.7+m)\ \mathrm{dB}\}$ with iterations of m based on 1% CDF level for DRC8 in ITU pedestrian A channel

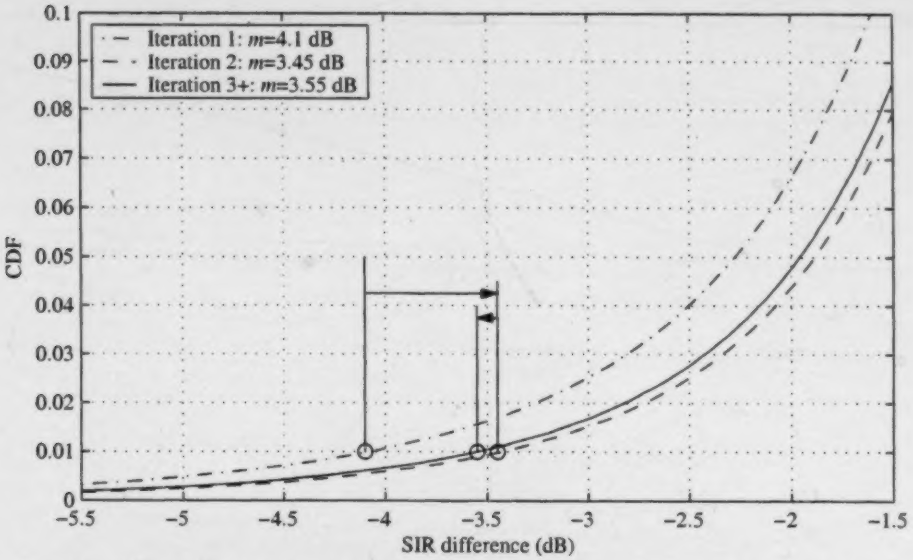


Figure 5.11: CDFs of $F\{\text{AVG}[\text{SIR}(slot+3),\text{SIR}(slot+7)]-\text{SIR}(slot) \mid \text{SIR}(slot) > (1.5+m)\text{ dB}\}$ with iterations of m based on 1% CDF level for DRC9 in ITU pedestrian A channel

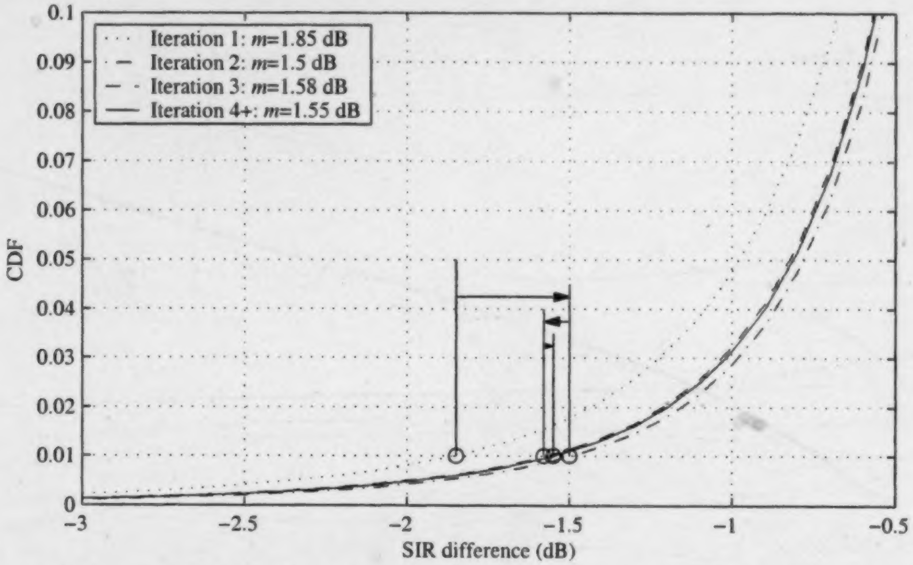


Figure 5.12: CDFs of $F\{\text{SIR}(slot+3)-\text{SIR}(slot) \mid \text{SIR}(slot) > (7.1+m)\text{ dB}\}$ with iterations of m based on 1% CDF level for DRC10 in ITU pedestrian A channel

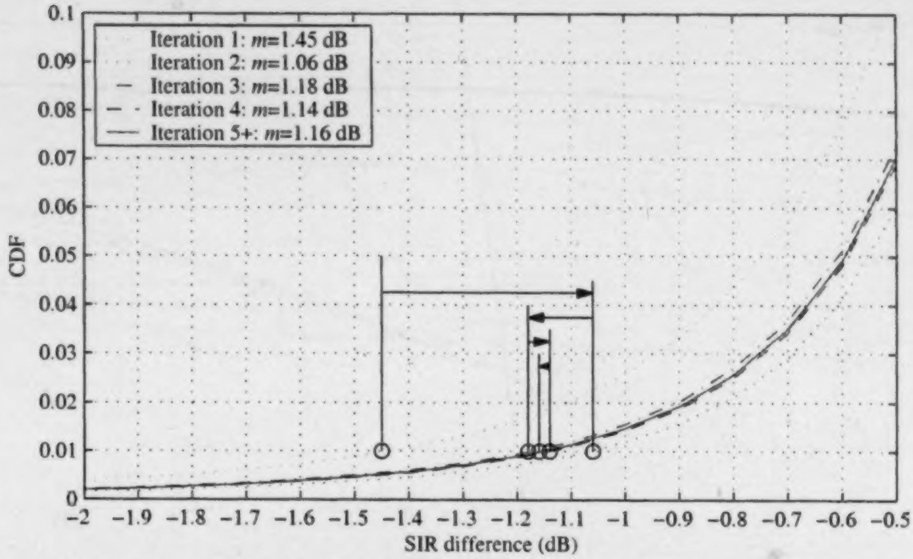


Figure 5.13: CDFs of $F\{\mathrm{SIR}(slot{+}3)-\mathrm{SIR}(slot) \mid \mathrm{SIR}(slot) > (9.2+m)\ \mathrm{dB}\}$ with iterations of m based on 1% CDF level for DRC12 in ITU pedestrian A channel

Based on the 1% levels on the CDFs, the margins for the ITU pedestrian A channel (after 5 iterations) should be as given in the following table:

TABLE IX:
ITERATIVE MARGINS PER DRC BASED ON 1% LEVEL FROM SIR CDFS

Format	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
DRC1	7.25	–	–	–	–
DRC2	6.7	6.65	–	–	–
DRC3	6.05	5.8	–	–	–
DRC4	4.8	4.2	4.3	–	–
DRC5	2.9	2.55	2.6	–	–
DRC8	2.35	2.01	2.06	–	–
DRC9	4.1	3.45	3.55	–	–
DRC10	1.85	1.5	1.58	1.55	–
DRC12	1.45	1.06	1.18	1.14	1.16

Using the margins in the above table, the RR, maxD, and PF algorithms were run in the ITU pedestrian A channel. Unfortunately, the resulting PERs per user wound up being far too low, except in the case of 2 users per sector. The PERs are shown in the figure below.

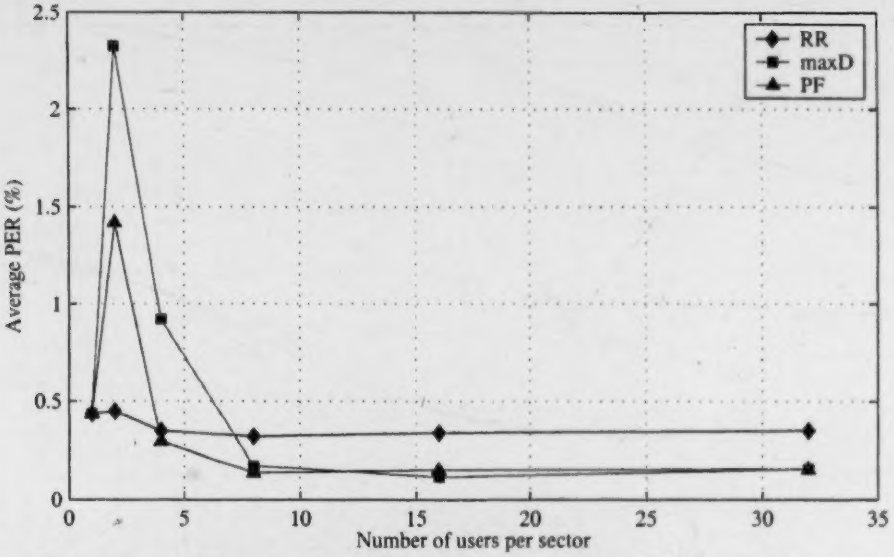


Figure 5.14: Average PER per user vs. number of users per sector for ITU pedestrian A channel using margins obtained from 1% level on SIR difference CDF curves

Since the PERs were too low in most cases, clearly the margins had to be reduced. The iteration process was re-run, this time targeting the 5% level on the CDFs instead of the 1% level. In this case, only 3 iterations were required to achieve a steady-state value for all the margins. These iterations are shown in the figures and table below.

TABLE X:
ITERATIVE MARGINS PER DRC BASED ON 5% LEVEL FROM SIR CDFS

Format	Iteration 1	Iteration 2	Iteration 3
DRC1	5.26	5.33	–
DRC2	5.1	–	–
DRC3	4.25	4.07	–
DRC4	2.67	2.5	2.52
DRC5	1.5	1.44	–
DRC8	1.24	1.15	1.16
DRC9	2.28	2.08	2.1
DRC10	0.98	0.88	0.89
DRC12	0.75	0.66	0.67

Using these new margins, the RR, maxD, and PF algorithms were again run in the pedestrian A channel. Unfortunately, the PERs per user still were not at the desired 1% level, as shown in the figure below.

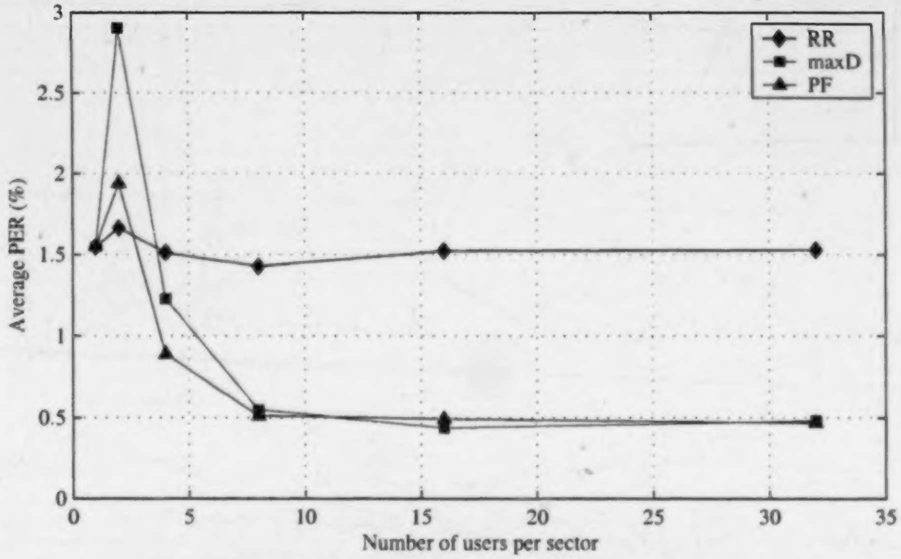


Figure 5.15: Average PER per user vs. number of users per sector for ITU pedestrian A channel using margins obtained from 5% level on SIR difference CDF curves

At this point, two possible directions for margin adjustment came to mind. The first was to take the margins using one of the CDF methods (i.e. the 1%-level margins), and add a constant bias to all the margins such that the average PER per user became 1%. This bias would depend on the scheduling algorithm and the number of users per sector. The other method be, rather than targeting a 1% PER per user, instead to target a 1% PER per format. That is, the margins would be adjusted such that out of all packets sent by all users with a given format, about 1% of those packets would be in error. In most cases, since the average packet error rate across all packets is about 1%, the average PER per user should also be about 1%. The adjustment of the margins would be on an iterative basis, which leaves the question of what point to start the iterations from. Initially, it was thought that the 1%-level margins would be a good starting point. However, upon further reflection, it was decided that starting the iterations with all margins equal to zero would be a better idea. One of the goals for the system design in most cases is to maximize the system throughput. In order to do so, the system must transfer data to users at the highest possible rate for each user. If the margins are not set properly (i.e., they are too large), the data rates requested by the users will not in fact be the highest rate they can support. It is feasible that there could be several sets of margins for which the average PER per format will be 1%. The set that should be used should be the one with the smallest margins, as that would be the set which maximizes the throughput. Thus, it makes sense to start the margins at zero, and increase them until a 1% PER is reached.

Note that it is possible that a 1% PER per format might not be supportable in some cases. In this case, the margin would increase until that format is effectively cut off. That is, the margin for the given format is high enough that the difference of (SIR – margin) would always meet the threshold of a higher rate before the 1% PER level for the given format is reached. As an example, consider formats DRC1, DRC2, and DRC3, with 1% PER threshold values $SIR_{1\%}$ of –13.5 dB, –10.5 dB, and –7.4 dB, respectively. Furthermore, assume that DRC1 and DRC3 achieve a 1% PER with a margin of 1 dB in both cases. This means that the measured SIR would have to be at least –12.5 dB for DRC1 to be requested, and at least –6.4 dB for DRC3 to be requested. Now, assume that the margin for DRC2 must be larger to achieve a 1% PER for that format. Assume further that the margin increases to 4.0 dB with the packet error rate for that format still being too high. By increasing the margin to 4.1 dB, this means that the SIR must be at least –6.4 dB for DRC2 to be requested. However, it is already known that –6.4 dB is sufficient for the higher rate DRC3 to be requested. Therefore, DRC2 is effectively cut off, as a SIR greater than –6.4 dB will cause DRC3 to be requested, and a SIR less than -6.4 dB will cause DRC1 to be requested, since DRC2 can't be supported at a PER of 1% at the lower SIR levels.

After comparing the two methods using the RR algorithm, it was found that the CDF-margins-plus-bias method yielded a somewhat higher throughput than the iterative margins method. However, the CDF method also displayed several drawbacks:

1. Although the average PER per user was the desired 1%, the average PER per format was not. In fact, the average PER for certain formats could be nowhere near the desired 1% level. The following table shows a comparison of packet error rates per format for the RR algorithm in the ITU pedestrian A channel.

TABLE XI:
AVERAGE PACKET ERROR RATES PER FORMAT USING RR ALGORITHM IN ITU PEDESTRIAN A CHANNEL USING VARIABLE MARGINS PER FORMAT PLUS BIAS TO OBTAIN 1% PER PER USER VS. USING ITERATIVE MARGINS PER FORMAT

Method / Format	4 users		8 users		16 users		32 users	
	Bias	Iterative	Bias	Iterative	Bias	Iterative	Bias	Iterative
DRC1	1.4846	0.9683	1.9299	0.9974	4.2198	0.9796	3.7794	0.9730
DRC2	1.2327	0.9668	1.1053	0.9973	2.1425	0.9529	1.5308	1.0406
DRC3	0.8044	1.0312	0.3135	1.0122	2.1425	0.9605	0.2641	0.9991
DRC4	0.5699	0.9991	0.1264	1.0406	0.1245	1.0284	0.1547	0.9744
DRC5	0.3074	1.0305	0.1122	0.9947	0.0992	0.9983	0.1200	1.0240
DRC8	0.4974	1.0235	0.2483	1.0177	0.1608	1.0237	0.1398	1.0057
DRC9	1.5922	N/A	0.6075	1.0118	0.2517	0.9922	0.1846	1.0024
DRC10	1.8602	1.0441	1.2649	1.0117	0.8269	1.0186	0.5746	1.0133
DRC12	1.7852	0.9991	2.0548	1.0093	1.8243	1.0186	1.8955	1.0127
Avg. PER per user	1.0202	1.0120	0.9972	1.0399	0.9791	1.0782	0.9907	1.0240

2. The margins to which the bias is applied would be dependent on the channel model. Ideally, the adjustment of the margins should be independent of the channel. This is true for the iterative margins method.

3. Using the bias method did not work for other algorithms in the case of 2 users per sector. Increasing the bias brought the average PER per user down to a certain minimum value greater than 1%, while further increases actually caused the PER to grow larger once more.

Because of these reasons, it was decided to use the iterative margins method to obtain the target PER per format. Starting at margins of zero, the 100-drop simulation would be run, and the error rates per format would be collected across all users and drops. The margins would then be adjusted accordingly based on the average PER per format, and the 100-drop simulation would be re-run. This process would repeat until all the PERs were sufficiently close to 1%. This then leaves the question of how close is "sufficiently close"? To begin with, it was decided that the resulting PERs should be within the range 0.95% to 1.05%, unless the number of packets received for a format was rather small (i.e. a few thousand) so that the PER values are less reliable. In this latter case, the range was expanded to 0.9% to 1.2 %. (The range is not symmetrical around 1%, since in most cases, the PERs are being reduced from higher values down into the range instead of from lower values up into the range.) In the event that only a few hundred packets or less were sent by a given format, it was decided to simply adjust the margins for those formats such that all the packets would be received correctly for a PER of 0%. This situation would be expected most often for the lowest rate formats, and due to the extremely low number of packets, the overall statistics would not be affected.

With the exception of the 0% situations as described above, in addition to being within the ranges given above, the 95% confidence interval for the PER values had to encompass the 1% value. The confidence interval was given in [39], under the condition that as the number of samples (i.e. packets) used to estimate the PER goes to infinity, the distribution of the packet error rates obtained from various trials of the experiment becomes Gaussian. In this event, for a given confidence probability 1–α (e.g. 1 – $\alpha = 0.95$, so that $\alpha = 0.05$), the range $y_- \leq p \leq y_+$ (where p is the observed PER) can be found by:

$$y_\pm = p\left\{1 + \left(\frac{d_\alpha^2}{2\eta}\right)\left[1 \pm \sqrt{\left(\frac{4\eta}{d_\alpha^2} + 1\right)}\right]\right\} \tag{5.1}$$

The value d_α in the above equation can be found by solving:

$$\frac{1}{\sqrt{2\pi}} \int_{-d_\alpha}^{d_\alpha} e^{-t^2/2} dt = 1 - \alpha \tag{5.2}$$

For $\alpha = 0.05$, $d_\alpha = 1.96$. The value $\eta = Np$, where N is the total number of packets sent by a given format and used to calculate the PER. Note that the equation to determine the confidence interval includes some approximations and is valid if $p(1\text{-}p) \approx p$ and $N/\left(N + d_\alpha^2\right) \approx 1$, both of which are true for the desired PER of 1% and N on the order of hundreds or larger.

The adjustment of the margins was determined to be complete if for all the formats used to transmit packets, the resulting PER per format fell within the aforementioned range, and the 95% confidence interval for the PER included the 1% level. This procedure was used when the current SIR less a margin was used for format determination, and also for the ITU vehicular B channel with perfect one-slot prediction. In the latter case, the margins for formats that only use one slot for transmission actually had negative margins. This was due to the fact that the PER curves drop off sharply with an increasing E_S/N_0. Since the prediction is perfect for 1-slot transmissions, and since the formats are not chosen unless the estimated SIR is above that required for a PER of 1%, this means that with a zero margin for those formats, the packet error rates are substantially below 1%, instead of being above. The negative margins have the effect of over-predicting the channel, so that the one-slot formats are requested at slightly lower SIR levels in order to bring the PERs up to the desired 1% range.

It was also desired to use this same procedure for the margins when using perfect multislot prediction. In this case, the margins required would have to be negative for all formats, and not just the one-slot formats as in the case of perfect one-slot prediction. However, using this method resulted in exceedingly long convergence times for the iteration process, as the resulting PERs were extremely sensitive to very small changes (e.g. 0.01 dB) of the margins. Therefore, for the perfect multislot prediction cases, it was decided to fix the margins at zero, and accept the lower PER values per format. In general, the resulting PERs were on the order of 0.01% to 0.1%. This likely did not affect the overall throughputs significantly, though. As a rough estimate, a 1% PER would likely have a throughput of 0.99 times that of a system with no errors. Similarly, for a system with a 0.05% PER, the throughput would be 0.9995 times that of a system with no errors. Therefore, a system with a 1% PER would on average have a throughput of $0.99/0.9995 \approx 0.99$ lower than the system with 0.05% PER. Overall, the 1% difference in throughput isn't very significant.

It should be noted that ideally, each user should have its own set of margins to adjust in order for each to obtain about a 1% PER per format. However, the run time of 18000 slots per drop does not allow for enough packets to be received by each user in all formats for such a method to be used. Therefore, the margins used in the simulation are global in nature and shared by all users requesting service per sector. This resulted in a distribution of PERs per user, but the average PER per user generally wound up being approximately 1%. In certain cases where a scheduling algorithm gave very few packets to some users in the sector, those PER for those users could be extremely high. For example, if a user received only 10 packets, and 1 was in error, the PER for that user would be 10%. This does not necessary reflect an actual PER due to the small number of packets, but the resulting larger percentages could skew the average PER per user to a value greater than 1%. When these sparse users were filtered out of the average, the average returned to being around 1%. Even without the filtering, the median PER per user almost always tended to be within 0.2% of 1%. Therefore, so long as the average PER per format met the 1% requirement, the median and filtered average PER per user were also around 1%, and so the other statistics in terms of throughputs and delays were also deemed to be usable.

Overall, the case of 2 users per sector generally resulted in somewhat higher values for the average PER per user than other cases, as evidenced by the graphs earlier this chapter. Aside from the small packet number explanation given above, the 2-user case also occasionally suffers from all users being deemed to be in outage. Take the example of the ITU pedestrian A channel. Overall, the margin required for DRC1 for almost all the scheduling algorithms were in the range of 4.5 to 5.5 dB. Taking an average of 5 dB, this means that a user would be considered to be in outage if its current SIR was less than (-13.5 + 5) = -8.5 dB. Looking at the CDF of SIR values (taking into account system imperfections) from the previous chapter, the probability of a user having a signal-to-noise ratio less than –8.5 dB is approximately 0.2. Therefore, since each user has an independent channel, the probability of two users being in outage simultaneously should be $0.2^2 = 0.04$. Common sense dictates that when both users are in outage, one must have a worse channel condition than the other. It is also likely that the worse user is likely to have been in that condition for some time. (This is particularly true for the ITU pedestrian A channel, which has flat fading, but is also true to a lesser extent for the other channels, which have multipath components.) Unfortunately, the scheduling algorithm has no direct knowledge about which of the two users has the worse channel condition. In most cases, since both users are considered equal, the algorithm chooses one at random. The rest of the time, the algorithm would select the user that has comparatively received the fewest resources to date. This would likely be the worse user. In either event, there is at least a 50% chance of the worst user of the two being selected to receive a packet using DRC1. The following figure shows the probability that, given the current SIR level, the average SIR over all slots where transmission of a DRC1 packet would occur will be below a -14.5 dB threshold. This value is used from the AWGN turbo code PER curves, as this level guarantees that the packet will be in error. Note that for higher SIR values, up to about 13 dB, there is still a probability that the packet will be in error, but the probability decreases rapidly due to the steepness of the turbo coding PER curves.

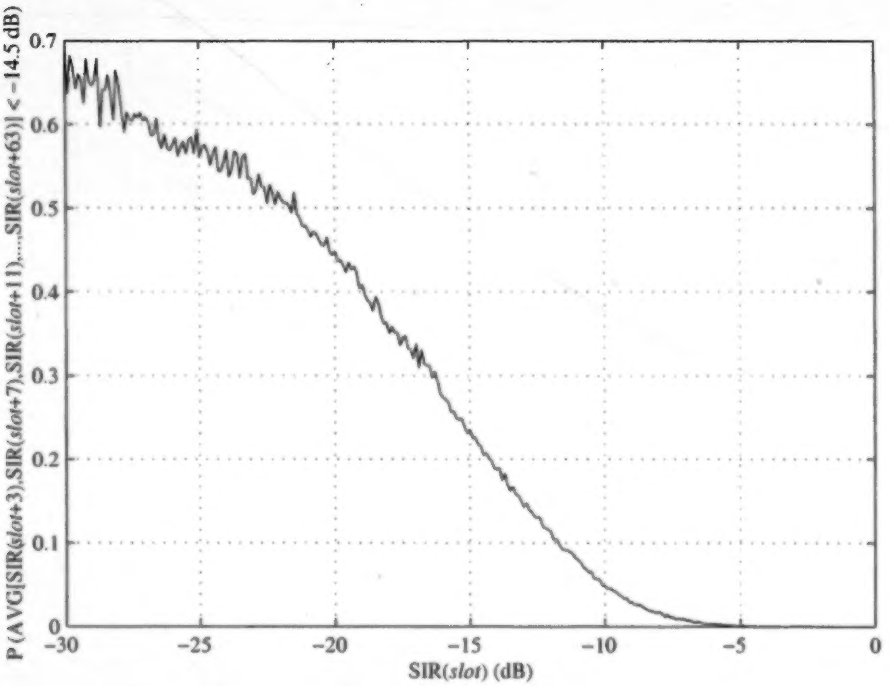


Figure 5.16: Approximate probability of a DRC1 packet error in the ITU pedestrian A channel, assuming that a DRC1 packet is scheduled to a user with an estimated signal-to-noise ratio of SIR(*slot*)

It can be seen that there is a significant probability of a packet error at the values of SIR where the worst user would likely be. In general, the average probability of a DRC1 packet error for a user deemed to be in outage can be found from the above error probability graph, in conjunction with the probability density function for the ITU pedestrian A channel SIRs, conditioned on the SIR being less than –8.5 dB. This PDF is shown below.

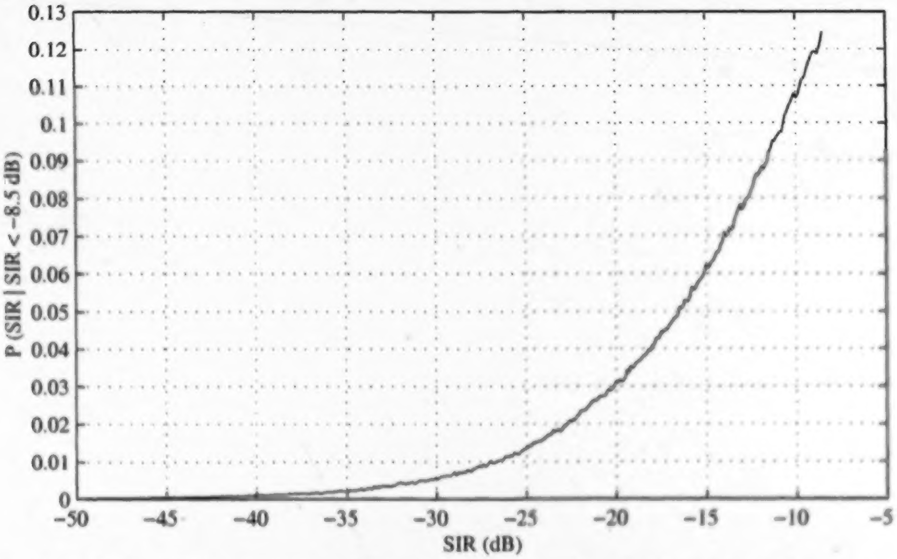


Figure 5.17: PDF of ITU pedestrian A SIR values, conditioned on the values being less than −8.5 dB

The average error probability for an outage user can be found by:

$$E[Error\_prob \mid S < -8.5 \text{ dB}] = \int Error\_prob(S) \cdot P[S \mid S < -8.5 \text{ dB}]\, dS \quad (5.3)$$

In the above equation, the SIR has been denoted by S for ease of notation. Calculating the above integral, it can be found that the average packet error probability for an outage user is 0.2354. Therefore, for the 4% of the time that both users are in outage, the probability of a packet error occurring is 23.54%. This results in higher-than-normal packet error rates for at least one of the two users, especially if the outage lasts for an extended period of time. The result is that the average PER per user increases, although the average PER per format remains at about 1%. This effect is only seen for 2 users per sector. For larger numbers of users, the probability of all of them being in outage simultaneously is negligible. In the case of 1 user per sector, the adjustment of the margins has the effect that any additional errors caused by outage transmissions are offset by a lower probability of error for non-outage transmissions. In addition, because there is only one user, it is likely that 4 packets with DRC1 will be transmitted in parallel, thereby tying up the channel for 64 slots, which allows the user time to exit the outage condition. These two effects mean that the 1 user per sector case is not affected in the same way as two users per sector.

With a method to adjust the margins available, useful results from the simulations can now be obtained. These begin in the following section, with an analysis of the performance of the scheduling algorithms in the pedestrian channels, based on the current SIR estimate and the tuned margins.

5.3. Pedestrian Channels, No Prediction (i.e. Using Current Estimated SIRs + Margins)

5.3.1. Round Robin (RR), Maximum Rate/DRC (maxD), and Proportionally Fair (PF A & B)

The round robin, maximum rate, and proportionally fair algorithms are probably the most well known scheduling algorithms available for the 1xEV-DO system. Hence, it makes sense to analyze these algorithms first.

The following figures show the average sector throughput per 30-second drop as a function of the number of users per sector for the ITU pedestrian A and B channels.

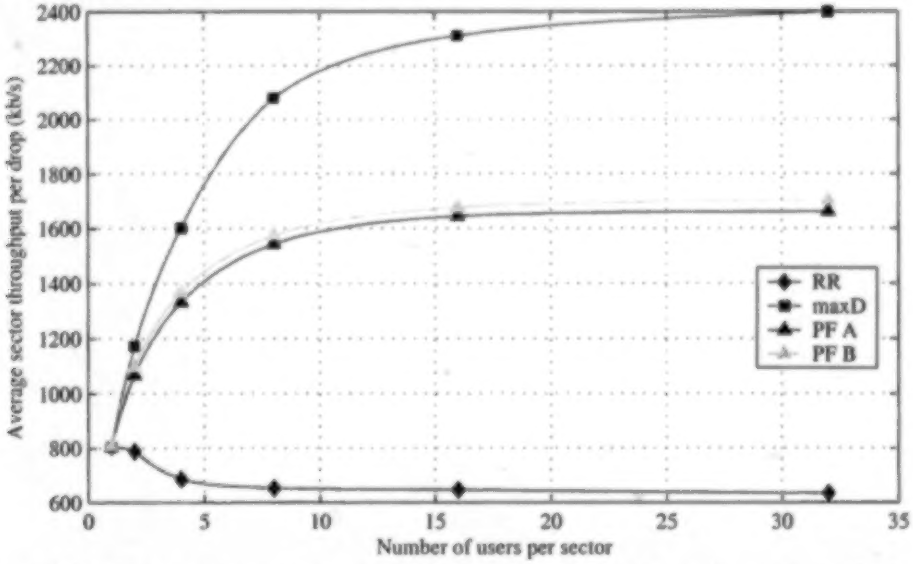


Figure 5.18: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian A channel for the round robin, maximum rate, and proportionally fair scheduling algorithms

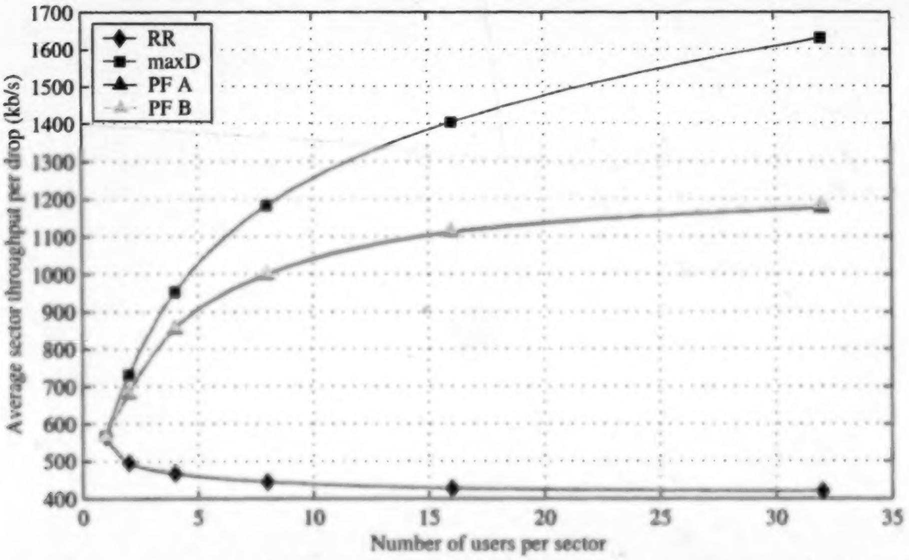


Figure 5.19: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian B channel for the round robin, maximum rate, and proportionally fair scheduling algorithms

As expected, the RR and maxD algorithms provide a very low and a very high throughput respectively, while the PF algorithm yields a throughput towards the middle of this range. There is very little change to be seen in the throughput due to the different methods of calculating the average rate between the PF A and PF B algorithms. The largest difference between the two is in the case of 2 users in the ITU pedestrian A channel, which displays an increase in throughput of only 1.7%, which is negligible.

In terms of multiuser throughput gain, the maxD algorithm shows an increase in sector throughput from about 800 kb/s to about 2400 kb/s in the ITU pedestrian A channel and from about 565 kb/s to about 1630 kb/s in the ITU pedestrian B channel. This corresponds to a gain of 3 and 2.885, respectively. Similarly, the PF algorithm shows an increase from 800 kb/s to 1680 kb/s and 565 kb/s to 1175 kb/s respectively, for a gain of 2.1 and 2.08 for the respective channels.

It is interesting to note that the throughput drops with increasing sector population for the round robin algorithm. This at first may seem surprising, as one might expect the throughput to be constant for all numbers of users. The reason behind the drop lies in the fact that users are distributed uniformly over the area of the sector, not uniformly over a distance from the transmitter. Consider an extremely simple case of a square-shaped sector, in which users are uniformly distributed. Assume that users are dropped in groups of 1, 4, 9, and 16 in the sector. On average, the distribution of the users would be as shown in the figure below.

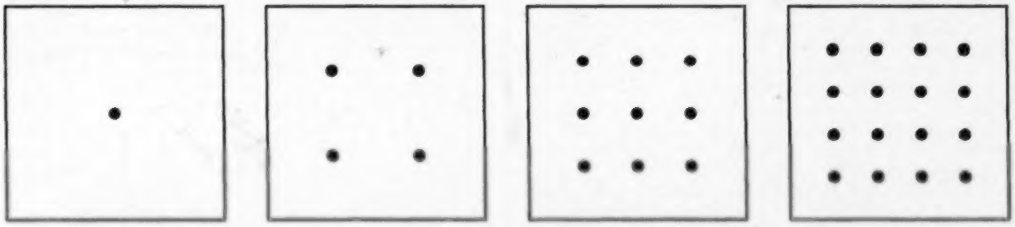


Figure 5.20: Square sectors with groups of 1, 4, 9, and 16 users distributed uniformly over the area of the sector

As the number of users in the sector increases, there is on average a higher proportion of the total number of users farther away from the transmitter, regardless of its location in the sector. This increase is particularly apparent when the transmitter is located at the corner of the sector, as it is in the case of the radio environment model for the simulations. Therefore, based purely on path loss, as the number of users in a sector increases, there will on average be more users requesting lower data rates. A more formal demonstration of the fact that any given proportion of the users is further away from the transmitter as the number of users per sector increases is shown for a hexagonal-shaped sector in Appendix K.

Hence, because the round robin algorithm does not consider channel conditions or requested rates in the scheduling decision, the average sector throughput per drop decreases with increasing users per sector. The throughput decreases from 1 to 32 users per sector by a factor of 0.8 and 0.74 for the ITU pedestrian A and B channels, respectively.

The following two figures show cumulative distributions of the average throughput per user per drop for the two pedestrian channels in the case of 16 users per sector.

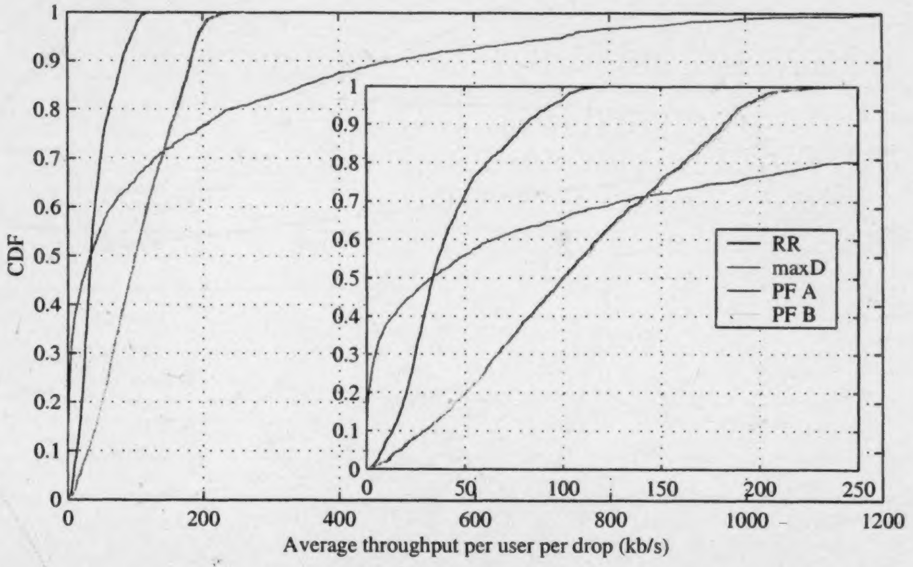


Figure 5.21: Distributions of average throughput per user per drop for the round robin, maximum rate, and proportionally fair algorithms in the ITU pedestrian A channel with 16 users per sector

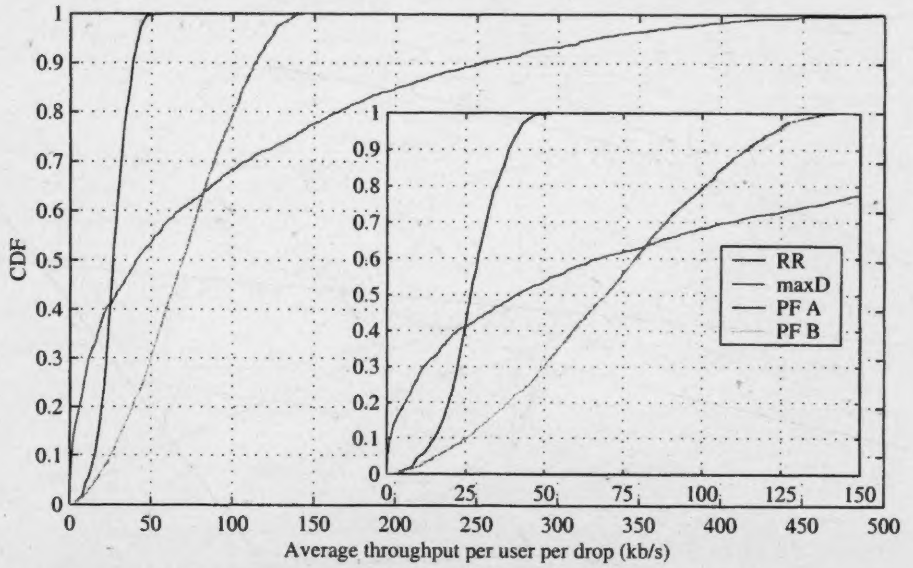


Figure 5.22: Distributions of average throughput per user per drop for the round robin, maximum rate, and proportionally fair algorithms in the ITU pedestrian B channel with 16 users per sector

The following table shows the mean, median, and standard deviation of the average throughput per user per drop distributions.

TABLE XII:
STATISTICS FOR AVERAGE THROUGHPUT PER USER PER DROP DISTRIBUTIONS FOR THE ROUND ROBIN, MAXIMUM RATE, AND PROPORTIONALLY FAIR SCHEDULING ALGORITHMS IN THE ITU PEDESTRIAN A AND B CHANNELS WITH 16 USERS PER SECTOR

Channel	ITU pedestrian A				ITU pedestrian B			
Algorithm	RR	maxD	PF A	PF B	RR	maxD	PF A	PF B

Mean (kb/s)	40.7	145.7	103.6	104.7	26.7	87.7	69.5	69.7
Median (kb/s)	33.7	33.4	99.6	100.6	26.5	41.0	69.6	69.8
Std. Dev. (kb/s)	25.6	236.8	55.6	56.7	8.95	106.5	32.5	32.8

It can be seen that the round robin algorithm has the tightest distribution of throughputs of the compared algorithms. That is, there is the smallest deviation between the throughputs per drop achieved by all the users in the sector. However, the actual throughputs themselves are also comparatively small. This is a result of the RR algorithm not taking channel conditions into account in its scheduling decisions.

The maximum rate algorithm displays some of the highest throughputs per user. However, it simultaneously displays a high probability of extremely low throughputs. The probability of a user not receiving any data during a drop (i.e. being starved for service) is 12.3% in the ITU pedestrian A channel and 1.9% in the ITU pedestrian B channel. 20.1% and 9.5% of the time, a user will receive a throughput less than 1 kb/s for the two channels, respectively. The large standard deviation values and the notable difference between the mean and median values are further indications of how varied the throughputs achieved by each user can be. Therefore, while the maximum rate algorithm can achieve very high throughputs, it is also extremely unfair in terms of the amount of data each user receives.

The proportionally fair algorithm seems to achieve a reasonable tradeoff between throughput and fairness. The distribution of throughputs is still fairly tight, but the actual throughputs themselves are much higher compared to the RR algorithm. There is also very little difference to be seen in terms of the PF A and PF B versions of the proportionally fair algorithm.

Note that the standard deviations of throughputs are lower in the ITU pedestrian B channel than in the ITU pedestrian A channel. While this is partly due to the fact that the throughputs themselves are lower, it is also due to the fact that the B channel displays more variability in the signal strength due to multipath. The more frequent changes in SIR mean that the user who is relatively best will also change more often. As a result, the differences seen in throughput between users are smaller. Also of interest is the fact that the median throughput per user tends to be somewhat lower than the mean in the ITU pedestrian A channel, whereas they are much closer together in the ITU pedestrian B channel. This again is a result of the signal strength variability of the channels. In the case of the ITU pedestrian A channel, the slow changes in the signal strength mean that if a user receives a weak signal, the signal will remain poor for an extended period of time, much longer than compared to the ITU pedestrian B channel. This would therefore mean that if a user were in outage, that user would remain in outage for a significant length of time. The throughput achieved by the few users in this case would lower the median throughput more than they would the mean throughput, as the higher proportion of lower throughput users would shift the median to the lower end of the scale. In comparison, the mean may actually increase somewhat, as during the time the outage user cannot be served, other users would be served, thereby increasing their throughputs. Thus, in the ITU pedestrian A channel, the medians can be expected to be lower than the means, in comparison to the ITU pedestrian B channel, where the median and mean throughput per user per drop are much closer in value.

The following two figures show the proportions of packets transmitted under a given scheduling algorithm achieving an effective bit rate, in the case of 16 users per sector. The effective bit rates are the actual bit rates with which a packet is received, taking into account early acknowledgements. For example, a packet transmitted using the format DRC3 (nominally 153.6 kb/s and 4 slots) that is received correctly using only 3 slots would have an effective bit rate of 204.8 kb/s.

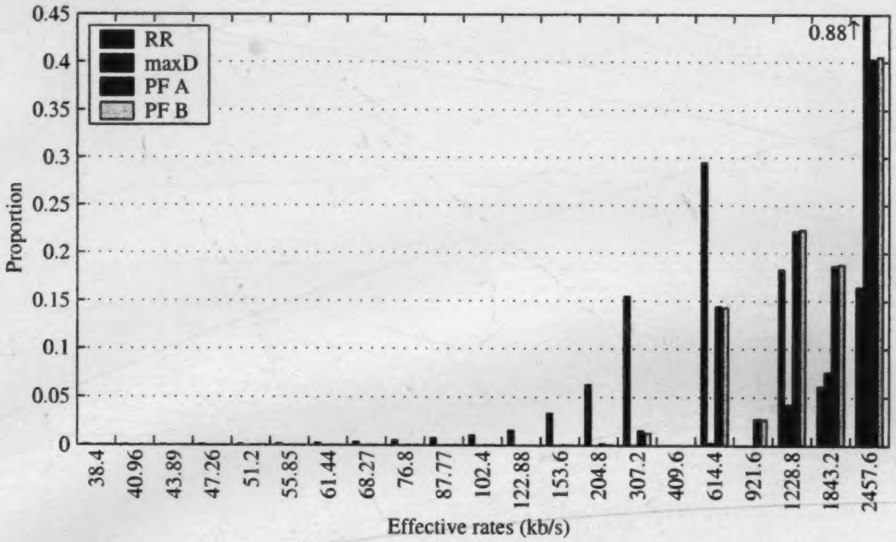


Figure 5.23: Distributions of effective bit rates for packets transmitted under the round robin, maximum rate, and proportionally fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector

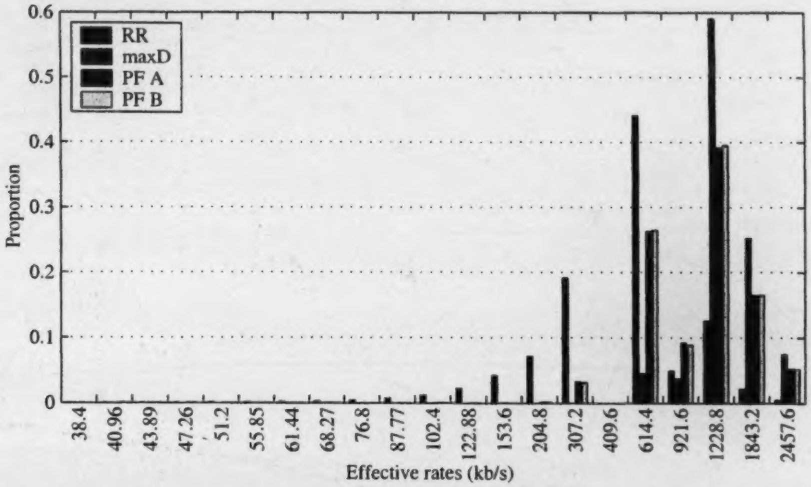


Figure 5.24: Distributions of effective bit rates for packets transmitted under the round robin, maximum rate, and proportionally fair scheduling algorithms in the ITU pedestrian B channel with 16 users per sector

To further illustrate how these rates were achieved, the following two diagrams give the distributions of the formats used for correctly received packets, and the number of slots used to transmit each of those packets. As before, the case of 16 users per sector is used.

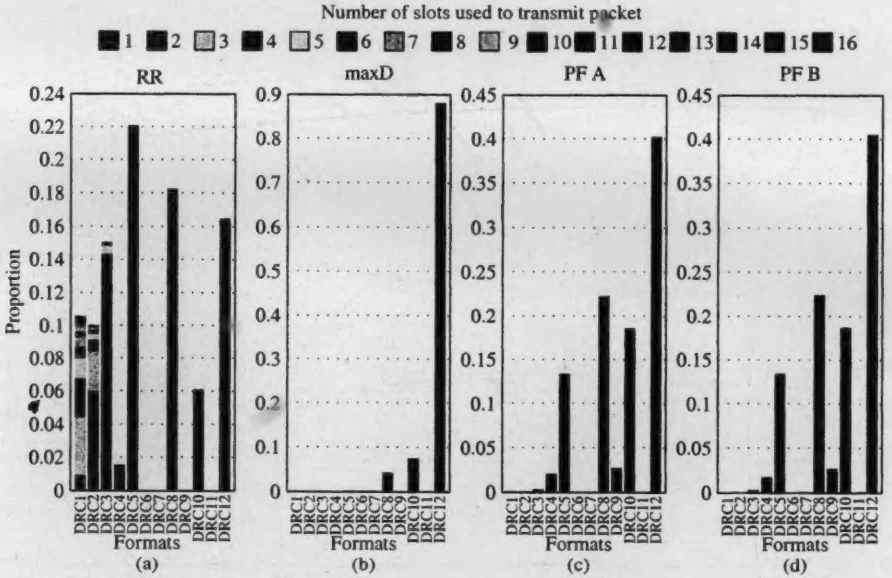


Figure 5.25: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a) round robin, (b) maximum rate, and (c),(d) proportionally fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector

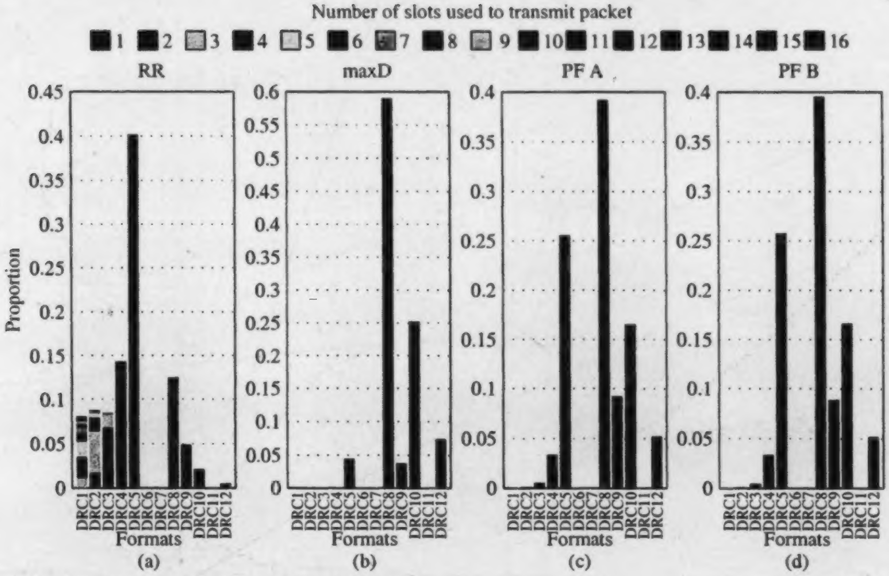


Figure 5.26: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a) round robin, (b) maximum rate, and (c),(d) proportionally fair scheduling algorithms in the ITU pedestrian B channel with 16 users per sector

It can be seen that other than with the round robin algorithm, most of the packets are transmitted using the higher data rates that are available. Even considering the round robin algorithm, most of the packets sent with the lower rate formats experience early terminations, thereby increasing their effective bit rates.

The fact that most of the packets are crowded around the highest rates under the maximum rate algorithm is no surprise. That is the design of the algorithm: to select the user with the highest rate. In the ITU pedestrian A channel, the highest rate user most of the time is requesting the highest rate format (2457.6 kb/s). In comparison, in the ITU pedestrian B channel, that highest rate format is most often DRC8 (1228.8 kb/s). This is likely due to the multipath nature of the B channel, as well as the fact that less of the total transmit power from the base station is captured, which means the highest rates cannot be supported as often as in the A channel.

There again is little difference to be seen between the A and B versions of the proportionally fair algorithm. Notice that there are almost no packets transmitted using the three lowest rate formats. Therefore, the nature of the proportionally fair algorithm to let users receive data when their instantaneous signal strength is greater than their average signal strength (or more accurately, when the rate the user requests is higher than the average rate they have been receiving data at in the past) means that in almost all cases, when the requested rate is sufficiently higher than the average rate (such that the user is scheduled), that rate will be at least 8 times higher than the minimum possible rate. This is yet another indication of how delaying packet transmissions can increase the resulting bit rates achieved.

The analysis of the formats used for correctly received packets also gives one possible reason why there is little difference to be seen between the two proportionally fair methods. Note that the vast majority of the packets transmitted use a format that only requires 1 slot. 95% of all packets in the ITU pedestrian A channel and 87% in the ITU pedestrian B channel use 1-slot formats under the proportionally fair algorithm. The difference between the PF A and PF B algorithms lies in how the calculation of the average rate takes into account early terminations. However, in most cases, there is no possibility of an early termination, since only 1 slot is used to transmit the packet. Therefore, there should be little difference expected in the resulting average rates, and hence not much difference in the overall results between the PF A and PF B algorithms. This would likely change in the ITU vehicular B channel, as in that case, the simulations specify that formats using multiple slots should be used over formats using fewer slots. This should likely increase the number of early terminations, and thus cause a discernable difference in the results between the PF A and PF B algorithms.

Thus far, the only results discussed have been in terms of throughputs and bit rates. In comparison, the next few sets of results will be in terms of delays. The following two diagrams show the average delay per packet per drop as a function of the number of users per sector.

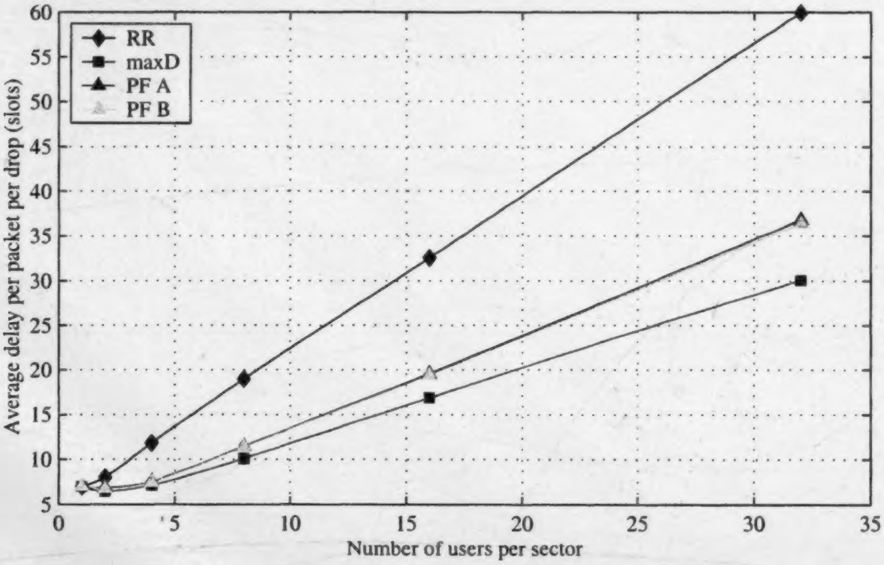


Figure 5.27: Average delay per packet per drop versus number of users per sector for the round robin, maximum rate, and proportionally fair scheduling algorithms in the ITU pedestrian A channel

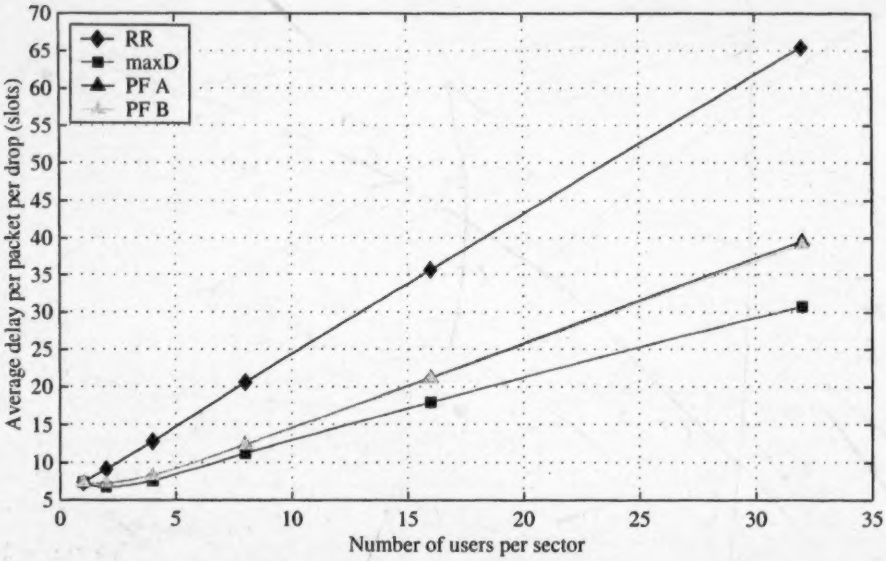


Figure 5.28: Average delay per packet per drop versus number of users per sector for the round robin, maximum rate, and proportionally fair scheduling algorithms in the ITU pedestrian B channel

The maximum rate algorithm provides the lowest average delay per packet, with the proportionally fair algorithm delay being somewhat higher. As before, there is almost no difference between the PF A and PF B algorithms. The round robin algorithm yields the highest average delay per packet, which at first glance may seem somewhat surprising, since all users are treated equally in terms of the number of packets they receive. However, this result is rather simple to explain. Consider a user in the system who can only support the lowest possible rate of 38.4 kb/s. If that user is selected for transmission, the delay seen by the packet will be 64 slots (16 transmissions × 4 slots in between transmission attempts), plus whatever head-of-line delay the packet has already experienced. During the transmission, those 16 slots are reserved for that user, so no other user could use the channel during that time, resulting in the delays for the other packets increasing. Now, consider an algorithm that takes into account channel conditions. Seeing that the given user can only support 38.4 kb/s, the algorithm may choose to delay transmission to that user in favor of other users that support higher rates, probably using formats that take fewer slots to transmit. Say that the transmission of the given user is delayed for 32 slots. At that point in time, the user's channel condition may have improved such that 76.8 kb/s can be supported. If a packet is scheduled to that user at this time, the overall delay for that packet will be no greater than in the original case where the transmission was scheduled earlier at 38.4 kb/s. However, because the transmission was delayed, other users were able to use the channel, thereby reducing the resulting delays for those packets. Hence, overall, the average packet delay seen by all packets during that period of time was reduced. Therefore, it is reasonable to expect that the average delay seen by packets in a system with a round robin scheduler will be higher than that seen in a system where channel conditions are taken into account.

To compare with the average delay per packet, the following figures show the average packet delay per user per drop as a function of the number of users per sector. Both a linear and a logarithmic scale are shown for each of the pedestrian channels.

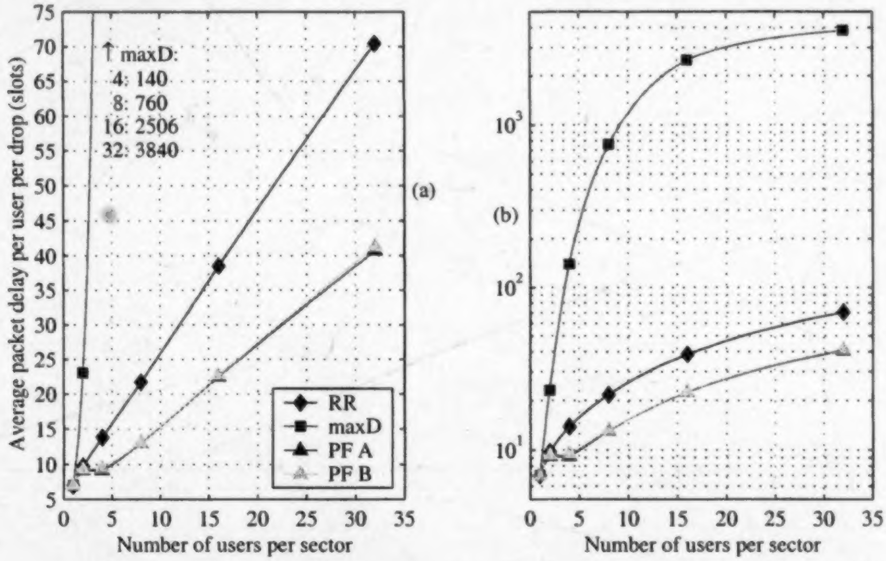


Figure 5.29: Average packet delay per user per drop versus number of users per sector for the round robin, maximum rate, and proportionally fair scheduling algorithms in the ITU pedestrian A channel on a (a) linear scale and (b) logarithmic scale

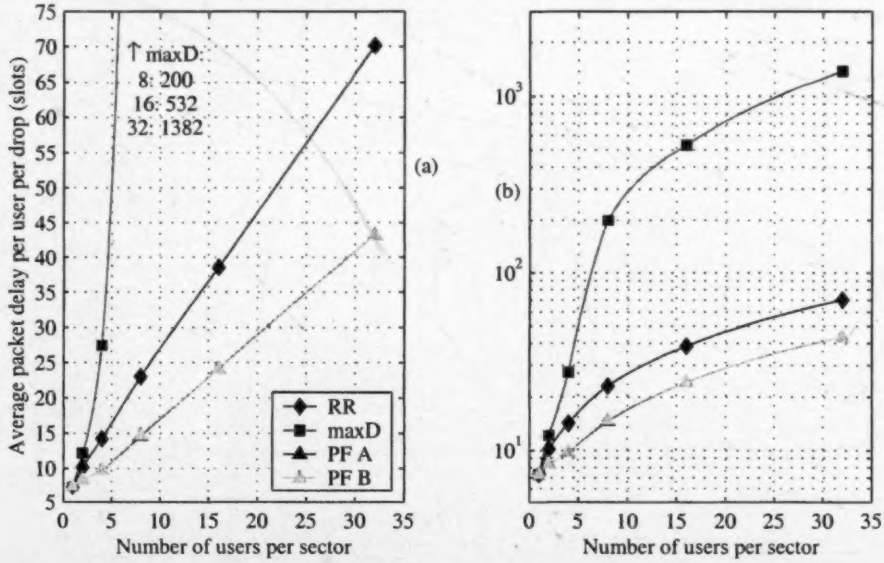


Figure 5.30: Average packet delay per user per drop versus number of users per sector for the round robin, maximum rate, and proportionally fair scheduling algorithms in the ITU pedestrian B channel on a (a) linear scale and (b) logarithmic scale

The average packet delay per user clearly demonstrates the unfairness of the maximum rate algorithm. While the average delay per packet was comparatively small for the maxD algorithm, in per user terms, the delay is extremely large, well over an order of magnitude larger than for the other algorithms. This is an effect of the maxD algorithm allocating most of the packet transmissions to a small group of users. While those few users receive many packets, and hence have a small average delay, most of the users receive very few packets, and consequently experience large delays on average. Therefore, when looking at the mean delay experienced by all users, the value is quite large.

For the round robin and proportionally fair algorithms, the average delays per user are much closer to their respective average delays per packet, indicating how much fairer they are than the maximum rate algorithm. However, the per-user delays are still about 10 to 20 percent higher than the per-packet delays, showing that there is still some variation in the number of packets each user receives. If every user in the system received the same number of packets, then the average delay per packet and average packet delay per user would be equivalent and equal.

As a final remark, note that there is still no discernable difference between the PF A and PF B algorithms.

The following graphs show the cumulative distribution functions (CDFs) of the delays experienced by all packets transmitted under a given scheduling algorithm with 16 users per sector. Note that these graphs only consider those packets for which the transmission has been completed (regardless of whether the packet was received correctly or not) by the end of a 30-second drop. Those packets whose transmissions have not been completed by the end of the drop, or that have not been scheduled for transmission before the end of the drop are not included.

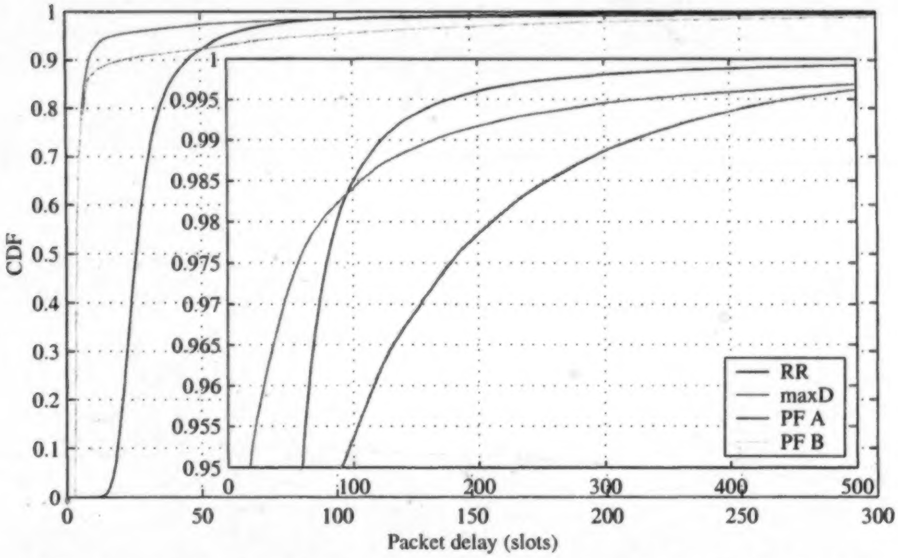


Figure 5.31: Cumulative distribution functions of delays experienced by packets under the round robin, maximum rate, and proportionally fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector

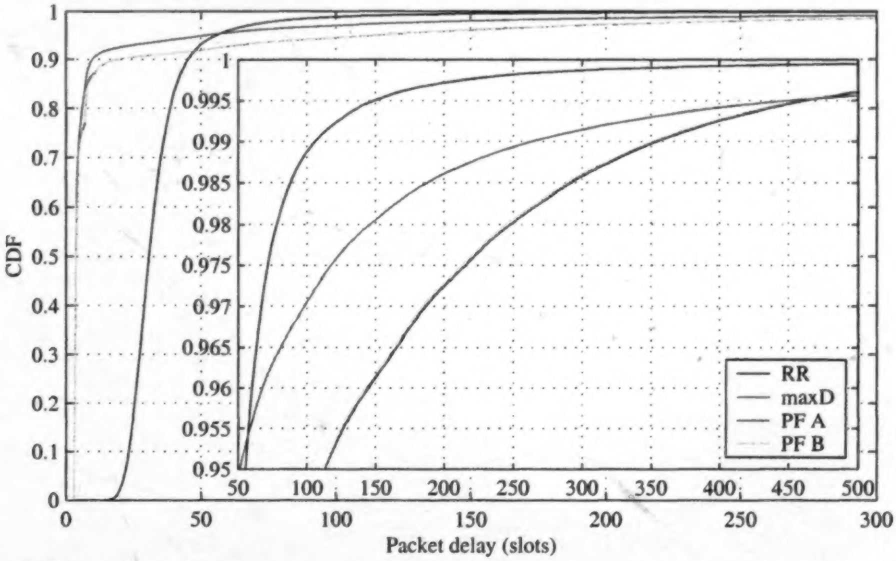


Figure 5.32: Cumulative distribution functions of delays experienced by packets under the round robin, maximum rate, and proportionally fair scheduling algorithms in the ITU pedestrian B channel with 16 users per sector

To better illustrate what happens at higher values of the distribution (i.e. at higher values of delay), the following graphs plot 1 minus the CDFs on a logarithmic scale. These graphs show the probability of a delay being greater than a certain number of slots, rather than the delay being less than that number of slots, as is the case above.

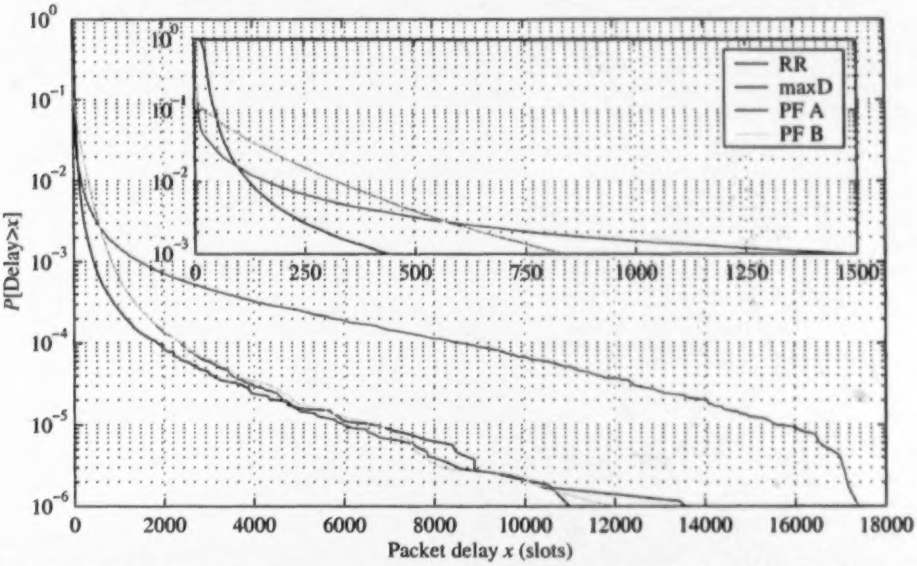


Figure 5.33: Probability of delays experienced by packets being larger than a given value under the round robin, maximum rate, and proportionally fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector

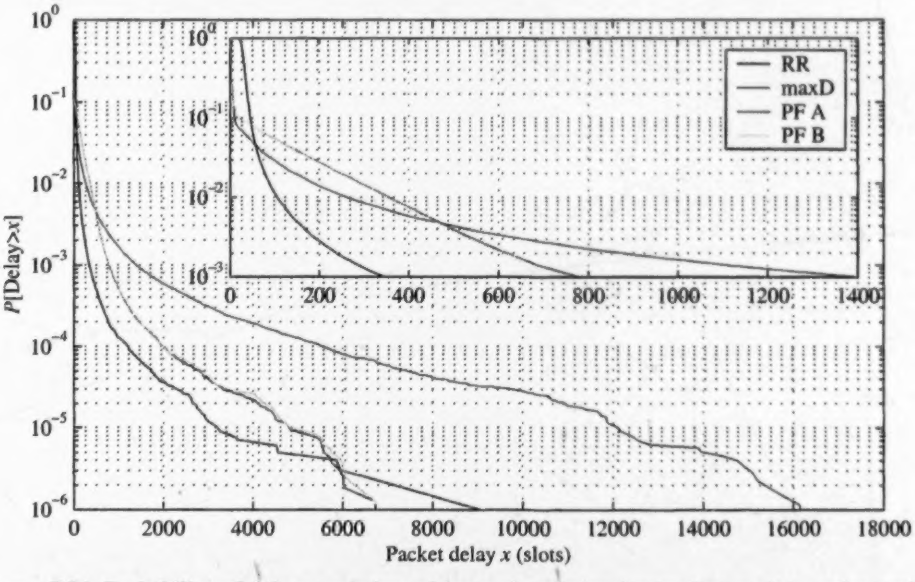


Figure 5.34: Probability of delays experienced by packets being larger than a given value under the round robin, maximum rate, and proportionally fair scheduling algorithms in the ITU pedestrian B channel with 16 users per sector

The maximum rate algorithm displays the highest probability of having packets with very low delays, yet at the same time has the greatest possibility of packets with very long delays. This is yet another indication of how the various users in a system operating under the maximum rate algorithm receive vastly different amounts of service. The packets with the lowest delays would correspond to the users with the best channel conditions, while those with the largest delays would correspond to the users with the worst channel conditions. (This statement is actually true for all the scheduling algorithms tested, but the difference in delays experienced is the most extreme under the maximum rate algorithm.)

The round robin algorithm shows the lowest probability of packets experiencing extremely large delays, primarily because each packet is guaranteed to be sent once the user's turn arrives in sequence (assuming the user is not in outage). Because of the cyclical nature of the scheduling decision, the variation in delays and the range of delays is also quite small.

The proportionally fair algorithm falls in the middle of the range between the round robin and maximum rate algorithms. The probability of a lengthy delay is somewhat larger than the round robin algorithm, but nowhere near as large as with the maximum rate algorithm. Once again, there is very little difference to be seen between the PF A and PF B versions of the proportionally fair algorithm.

The following table shows the various percentile delays for the scheduling algorithms in the two pedestrian channels. The p^{th} percentile delay is the maximum delay τ that $p\%$ of the packets transmitted under a given scheduling algorithm will experience. In other words, $p\%$ of the packets will experience a delay no greater than τ slots. The standard deviation in the delays is also shown at the bottom of the table.

Table XIII:
Percentiles and Standard Deviations of Packet Delays for the Round Robin, Maximum Rate, and Proportionally Fair Scheduling Algorithms in the ITU Pedestrian A and B channels with 16 Users per Sector

Channel	ITU pedestrian A				ITU pedestrian B			
Algorithm / Percentile	RR	maxD	PF A	PF B	RR	maxD	PF A	PF B
50th (median)	25.9	4	4	4	30.7	4	4	4
90th	45.2	8.9	20.9	20.3	45.1	9.9	19.6	19.4
95th	58.9	17.3	90.4	89.4	55.4	51.1	113.0	112.0
99th	123.4	164.7	322.6	319.8	105.8	263.2	353.6	350.2
99.5th	177.2	329.3	448.3	444.6	148.4	449.6	461.3	457.7
99.9th	438.9	1476.4	837.9	832.5	344.4	1399.0	777.2	768.5
99.95th	655.1	2835.4	1095.7	1092.0	493.2	2245.7	998.5	991.6
99.99th	1870.2	8673.6	2345.1	2350.6	1216.4	5549.0	1987.0	1972.3
Std. Dev.	52.06	168.80	80.64	80.84	30.10	132.28	76.93	76.73

The above table confirms how varied the delays are under the maxD algorithm and the tightness of the RR distribution. One very interesting observation is that except for the round robin algorithm, the median delay per packet is 4 slots. This means that at least half of the time, the packets sent under the maximum rate and proportionally fair algorithms have a delay of exactly 4 slots. The only way for a packet to obtain such a delay is for it to arrive at the head of the queue for a user, and then be immediately transmitted in the next slot. It can therefore be concluded that the majority of the time, the maxD and PF algorithms tend to send packets to users in groups; that is, a given user will receive several packets in succession before a new user is selected for transmission. This could possibly be an effect of the slowly changing nature of the pedestrian channels, such that the user who is currently "best" in terms of the scheduling algorithm metric tends to stay the "best" for a period of time. This effect does not occur in the round robin algorithm, since channel conditions are ignored when deciding which user should receive data, and since users are chosen cyclically.

The following two graphs show the distributions of the average packet delay experienced by each user under a given scheduling algorithm. Each user receives a different number of packets, and therefore has a different average delay seen across that group of packets, which leads to the distribution in question. The average packet delay per user seen earlier as a function of the number of users per sector is simply the average value of these distributions. As always, the case of 16 users per sector is considered, and similar to the packet delay distributions seen above, only the packets that have been transmitted by the end of each 30-second drop are considered for the distributions.

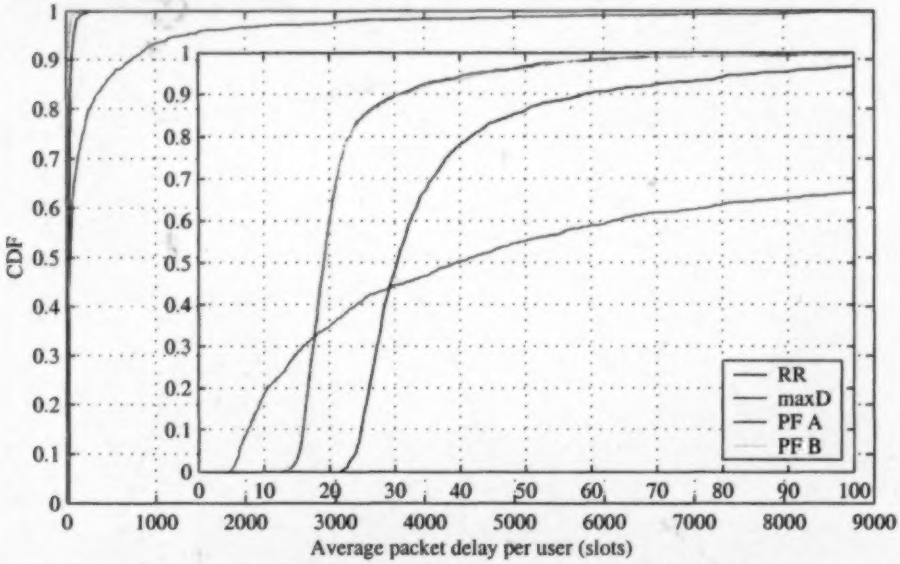


Figure 5.35: Distributions of average packet delay per user for round robin, maximum rate, and proportionally fair scheduling algorithms in ITU pedestrian A channel with 16 users per sector

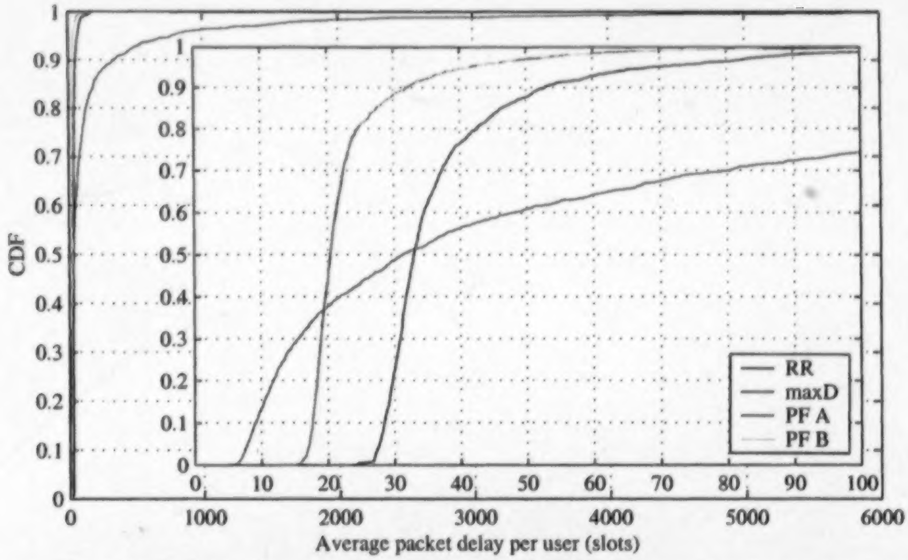


Figure 5.36: Distributions of average packet delay per user for round robin, maximum rate, and proportionally fair scheduling algorithms in ITU pedestrian B channel with 16 users per sector

The proportionally fair and round robin algorithms display a very narrow range of average delays per user. The vast majority of the time (i.e. at least 96%), the average delay that a user can expect for the packets it receives will be less than 100 slots (0.1666… seconds). The distributions of the delays are also quite steep, meaning that the variation of average delays between the users in the system will be small. For example, with the proportionally fair algorithm, approximately 80% of the time, the average packet delay a user will experience will be between 15 and 25 slots.

Users under the maximum rate algorithm experience a much wider range and generally much larger delays than with the other two algorithms. Consider the following table, which shows the standard deviation and various percentiles for the average delay per user for the two pedestrian channels.

TABLE XIV:
PERCENTILES AND STANDARD DEVIATION OF AVERAGE PACKET DELAY PER USER FOR THE ROUND ROBIN, MAXIMUM RATE, AND PROPORTIONALLY FAIR SCHEDULING ALGORITHMS IN THE ITU PEDESTRIAN A AND B CHANNELS WITH 16 USERS PER SECTOR

Channel	ITU pedestrian A				ITU pedestrian B			
Algorithm / Percentile	RR	maxD	PF A	PF B	RR	maxD	PF A	PF B
10th	24.8	7.5	15.9	15.7	28.4	9.1	17.7	17.6
25th	26.6	13.4	17.0	16.9	30.2	13.4	18.7	18.6
50th (median)	30.3	39.3	19.1	19.0	33.0	31.3	20.4	20.3
75th	37.8	186.4	21.8	21.7	39.0	102.7	23.2	23.2
90th	58.9	725.9	30.9	31.4	52.7	337.9	32.0	31.8
Std. Dev.	26.4	1102.1	14.1	15.4	17.7	710.5	21.9	22.2

The quick increase in the delays under the maximum rate algorithm is readily apparent when compared to the round robin and proportionally fair algorithms. There is a much larger probability of a user experiencing a long average delay in receiving its packets. The variations between the average delays seen by each user are also substantial.

As a final point of comparison, the following two graphs show the distributions of the allocated slots per user per drop. The slot allocation provides a view of how fairly the scheduling algorithm allocated system usage to the users on a time basis, and is somewhat secondary in importance to the throughput and delay results. 16 users per sector are assumed, so in a perfectly equal allocation of slots, each user would receive 1125 slots per 30-second (18000-slot) drop.

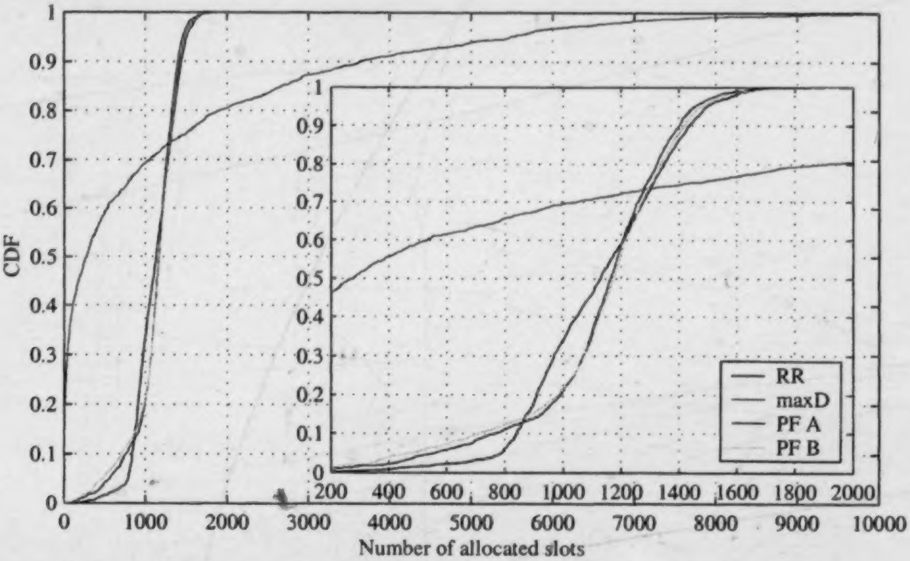


Figure 5.37: Distributions of allocated slots per user per drop for the round robin, maximum rate, and proportionally fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector

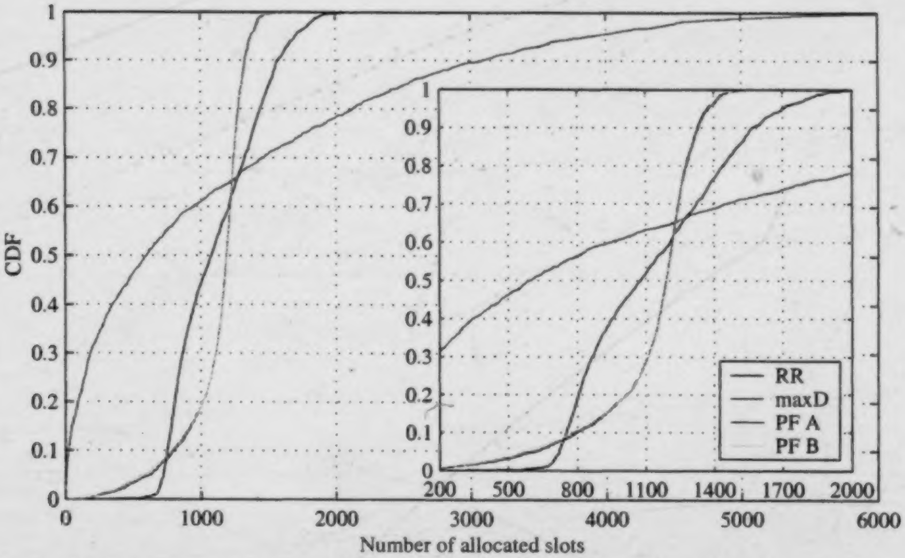


Figure 5.38: Distributions of allocated slots per user per drop for the round robin, maximum rate, and proportionally fair scheduling algorithms in the ITU pedestrian B channel with 16 users per sector

In the case of allocated slots per user per drop, the average value isn't particularly useful. If one were to sum up all slots allocated to all users, then divide by the number of users, the average would be 1125 slots in all cases. Rather, the tightness of the distributions and the percentile allocations are of more interest. These are shown in the following table.

TABLE XV:
PERCENTILES AND STANDARD DEVIATION OF SLOTS ALLOCATED PER USER PER DROP FOR THE ROUND ROBIN, MAXIMUM RATE, AND PROPORTIONALLY FAIR SCHEDULING ALGORITHMS IN THE ITU PEDESTRIAN A AND B CHANNELS WITH 16 USERS PER SECTOR

Channel	ITU pedestrian A				ITU pedestrian B			
Algorithm / Percentile	RR	maxD	PF A	PF B	RR	maxD	PF A	PF B
10th	840	0	766	715	752	21	797	785
25th	940	22	1040	1040	833	140	1075	1074
50th (median)	1130	275	1163	1173	1078	597	1190	1193
75th	1312	1433	1290	1309	1379	1764	1258	1262
90th	1436	3628	1401	1416	1571	3085	1328	1335
Std. Dev.	247.7	1794.6	265.9	290.5	324.4	1280.3	231.9	241.3

Both the round robin and proportionally fair algorithms provide a reasonably fair allocation of slots to all the users in the system. The proportionally fair algorithm tends to have a longer tail to the distribution on the lower end of the allocated slots scale, while the tail is somewhat longer for the round robin algorithm at the upper end. Note that the RR algorithm shows a smaller variation in the number of allocated slots for the ITU pedestrian A channel than the PF algorithm(s), whereas the reverse is true for the ITU pedestrian B channel. This is largely due to the more substantial variations in the signal due to multipath in the B channel, which allows each user to receive data more equally as they experience the relatively best conditions. In comparison to both these algorithms, the maximum rate algorithm shares the channel extremely unfairly.

Overall, the round robin algorithm has been shown to be reasonably fair to all users requesting service. However, that fact that it does not take channel conditions into account when making scheduling decisions means that the algorithm suffers both in terms of throughput and delay. The maximum rate algorithm provides an extremely high throughput per sector. The throughput per user can be quite high and the delays very low as well, but only for those few users who experience the best channel conditions. Otherwise, the throughput and delay seen by a user are extremely poor. Thus, the maximum rate algorithm, while providing the highest level of performance in some areas, also provides the highest level of unfairness.

The proportionally fair algorithm yields a good overall blend of throughput, delay, and fairness. While the throughput is nowhere near as high as with the maximum rate algorithm, the PF algorithm still provides over a 200% multiuser gain in per-sector throughput from 1 to 32 users, while still allowing all users to receive a reasonable amount of service. The PF algorithm also provides the best delay characteristics of the three algorithms, and is also extremely fair on a slot allocation basis.

There was no appreciable difference to be seen in the results due to the different methods of calculating the average rate between the PF A and PF B algorithms in either pedestrian channel. As mentioned earlier, one likely reason is the fact that the majority of transmissions occur with 1-slot format, negating the possibility of early terminations. Without early terminations, there is no difference between the two calculation methods. One other reason could be that in the cases where there are early terminations, the proportion that the average rate value increases for all users is approximately the same. This means that with the average rate in the denominator of the scheduling metric, the user who is relatively best would not change, as the metric for all the users would decrease by the same fraction. Therefore, any algorithm which involves directly multiplying or dividing by the average rate in the metric (i.e. the maxD/PF, M-LWDF 2, and EXP2 algorithms) will also not likely see any appreciable difference in the results in the pedestrian channels, and hence those algorithms will only be analyzed using the PF A method of calculating the average rate. (The PF A method is also the most well known in the literature, so if only one method is going to be examined for the other algorithms, it makes for it to be that method.) However, the latter reason might have an impact on the minperf scheduling algorithm, which involves a subtraction from the average rate instead of a multiplication or division. This could impact the relative value of the metric between users, and consequently the PF B method for the average rate calculation will be investigated for the minperf algorithm as well.

5.3.2. Maximum Rate / Proportionally Fair Hybrid (maxD/PF)

The main goal behind the design of this algorithm was to create a hybrid of the maximum rate and proportionally fair algorithms. The algorithm would have a tunable parameter that could control the throughput and delay characteristics between those provided by the two algorithms in question. More specifically, it was desired that the resulting average sector throughput per drop should be approximately linear with the tunable parameter between the two bounds of the constituent algorithms for all numbers of users per sector and for all channel models. To this end, the following metric was proposed:

$$a\frac{DRC_k(t)}{\bar{R}_k(t)}+(1-a)\frac{DRC_k(t)}{R_{denom}} \qquad (5.4)$$

The values $DRC_k(t)$ and $R_k(t)$ are the requested rate and average rate for user k in slot t, and are the same as defined for the proportionally fair algorithm. (The "PF A" version of the proportionally fair algorithm was used to calculate the average rates.) The value a is the tunable parameter. When a=0, the algorithm becomes equivalent to the maximum rate algorithm. Similarly, when a=1, the hybrid algorithm is equivalent to the proportionally fair algorithm. The value R_{denom} is a constant that is intended to normalize the value of the maximum rate portion of the above metric (i.e. the right half of the equation) to the range of that of the proportionally fair half of the metric (i.e. the left side). The user with the largest value for the overall metric would be the one selected to receive a packet.

Before the algorithm could be used, the proper value for R_{denom} first had to be found. A reasonable starting place seemed to be the average of the maximum and minimum possible transmission rates for 1xEV-DO. These are 2457.6 kb/s and 38.4 kb/s, respectively, so their average is 1248.0. This was the initial value used to test the algorithm. The following shows the resulting average throughput per sector per drop in the ITU pedestrian A channel with 16 users per sector. The parameter a was set at 0.25, 0.5, and 0.75. The "desired" throughputs are those that exactly split the throughput range between the maxD algorithm and the PF algorithm into quarters.

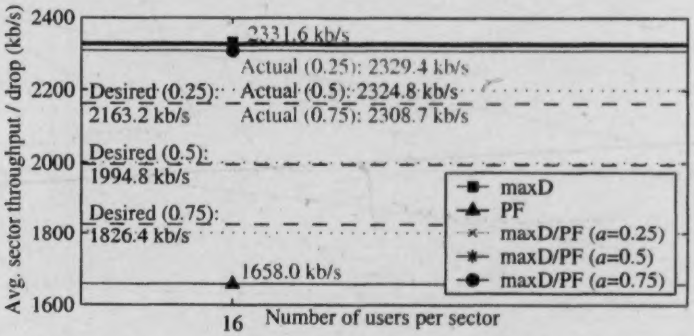


Figure 5.39: Average sector throughput per drop from maxD/PF scheduling algorithm in ITU pedestrian A channel with 16 users per sector and R_{denom}=1248

Obviously, the value of 1248 for R_{denom} was not appropriate. However, an error in the metric calculation was discovered upon closer examination. While the value of R_{denom} was obtained in units of kb/s, the remaining terms in the metric were being used in units of bits per second. Therefore, the value of 1248 was a factor of 1000 too low. The value of R_{denom} was corrected to 1248000 (in units of bits per second), and the simulations were rerun with the new constant. The resulting throughputs are shown below.

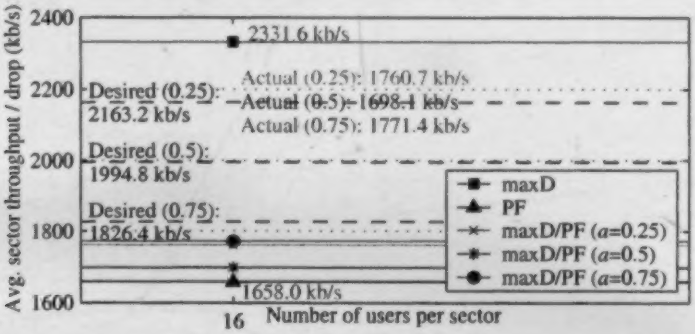


Figure 5.40: Average sector throughput per drop from maxD/PF scheduling algorithm in ITU pedestrian A channel with 16 users per sector and R_{denom}=1248000

Clearly, the value of 1248000 for R_{denom} also does not work. The values are grouped near the PF value, and strangely, the throughput decreases as *a* goes from 0.25 to 0.5, but then increases again as *a* goes from 0.5 to 0.75. The cause of this latter effect is uncertain, but in any event it is clear that the denominator must be changed.

Instead of picking a new value at random, it would be useful to investigate how the value of $\overline{R}_k(t)$ changes over time for various users under the proportionally fair algorithm. The following diagram shows typical $\overline{R}_k(t)$ values for a single drop of 16 users per sector in the ITU pedestrian A channel.

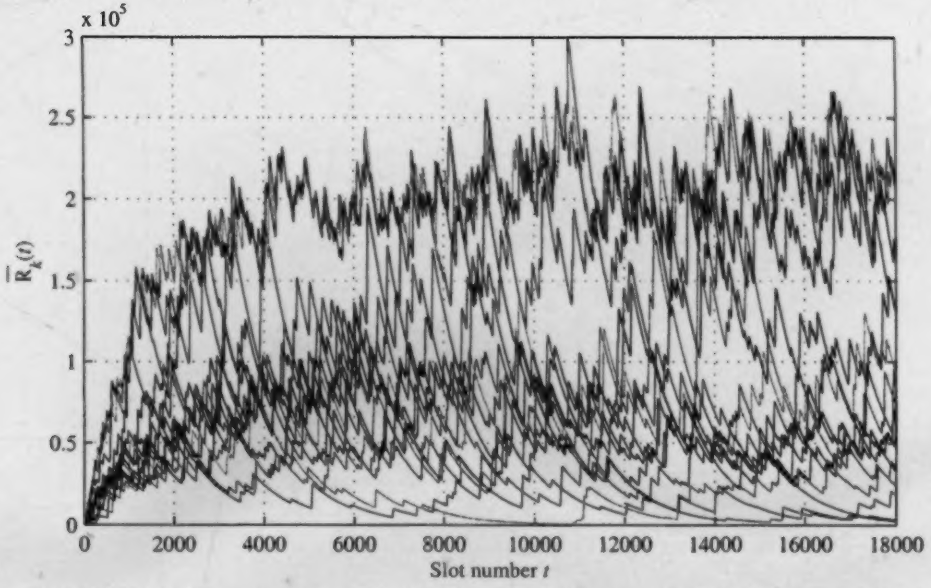


Figure 5.41: Typical average rate values (units of bits per second) in the proportionally fair scheduling algorithm metric for a drop of 16 users per sector in the ITU pedestrian A channel

In order for the maximum rate metric to be normalized to the range of the proportionally fair metric in the maxD/PF algorithm, the denominator should be the average of all the $\overline{R}_k(t)$ values. The following figure shows the mean of the average rate value for 160 users (16 users/drop × 10 drops) for each slot.

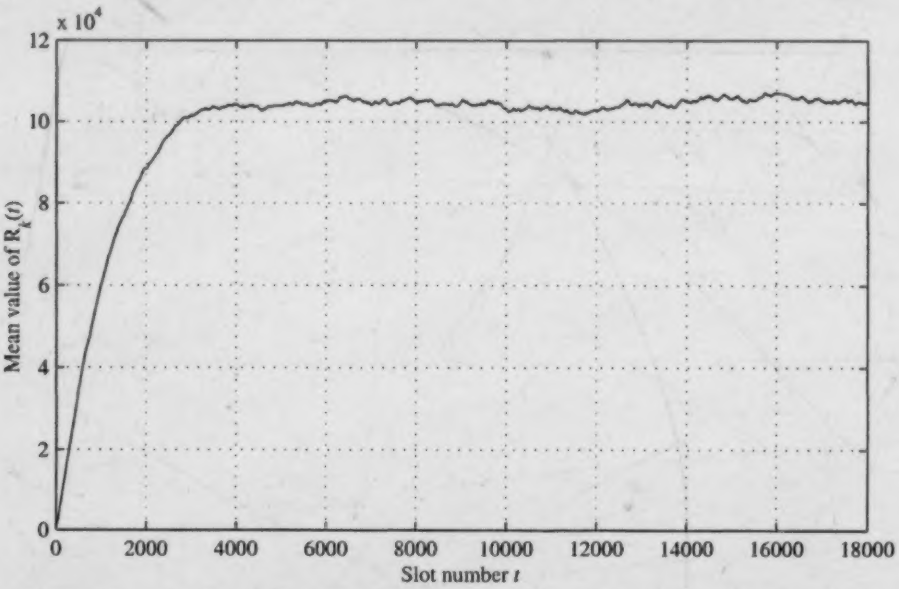


Figure 5.42: Mean of the average rate values in the proportionally fair scheduling algorithm across 160 users in the ITU pedestrian A channel

It can be seen that the average value of $\overline{R}_k(t)$ increases for approximately 3500 to 4000 slots, at which point it stabilizes to a value of about 105000 bits per second. This value was used for a new set of simulations of the maxD/PF algorithm. The resulting average throughputs per sector per drop are shown below.

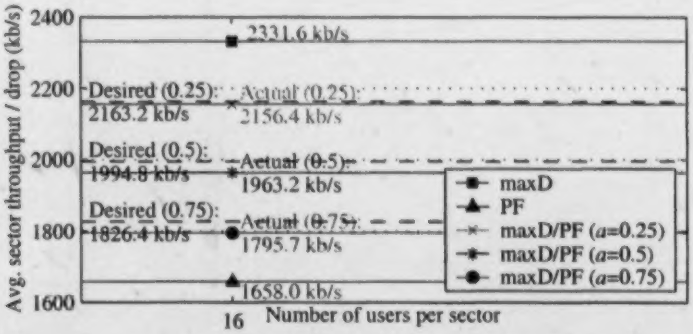


Figure 5.43: Average sector throughput per drop from maxD/PF scheduling algorithm in ITU pedestrian A channel with 16 users per sector and R_{denom}=105000

As can be seen from the above graph, the value of 105000 bits per second for R_{denom} works rather well in dividing the throughput range into quarters. In worst case (at a=0.75), the actual throughput differs from the desired throughput by only about 1.7%, which is not significant. Therefore, the test of the current denominator value can be extended to of numbers of users per sector. The following graph shows the average sector throughput per drop versus the number of users per sector for the maxD/PF algorithm with R_{denom} = 105000 bits per second in the ITU pedestrian A channel.

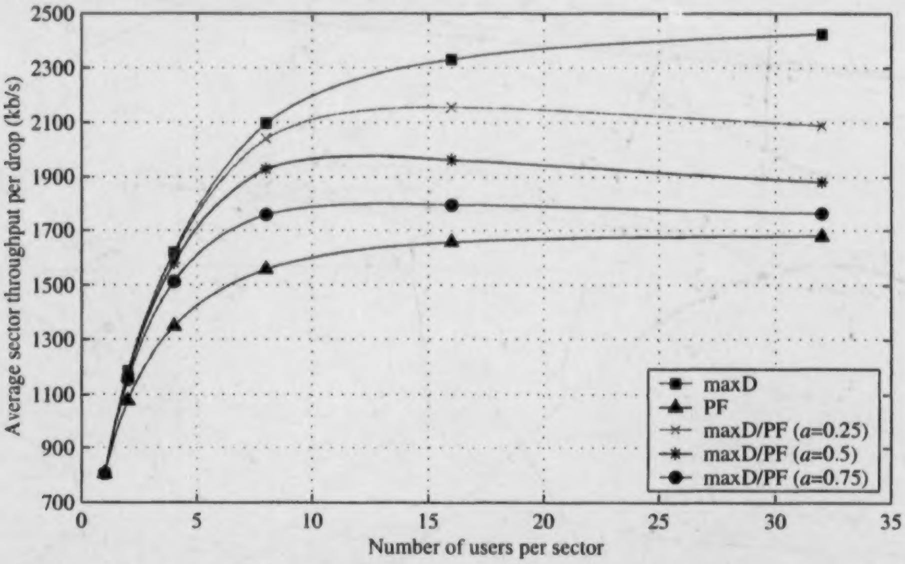


Figure 5.44: Average sector throughput per drop versus number of users per sector for maxD/PF scheduling algorithm with R_{denom} = 105000 bits per second in ITU pedestrian A channel

It is clear from the above figure that although the value of 105000 for R_{denom} linearizes the throughput with respect to the tuning parameter in the case of 16 users per sector, the same is not true for the other numbers of users per sector. For groups smaller than 16 users, the throughputs are biased towards the maximum rate algorithm, while above 16 users per sector, the throughputs are closer to the proportionally fair algorithm. This result is readily understandable. Consider the throughput of the maximum rate or proportionally fair algorithm. Although the average throughput per sector per drop increases with an increasing number of users, the average throughput per *user* per drop (which can be approximated by the sector throughput divided by the number of users per sector) decreases with increasing users. Thus, the average rate calculated for all users would also decrease. Hence, it would seem that there should be some sort of dependence of the appropriate value of R_{denom} on the number of users in the sector. The channel conditions would likely play a role in the value as well. A more severe multipath environment or faster fading would also likely decrease the throughput per user and therefore the average rate.

Ideally, we want the value of R_{denom} to either be immune to these various effects, which is not possible, or the value should adapt to the changing conditions, meaning the value is no longer a constant. Looking back at the equation for the maxD/PF algorithm metric, it can be surmised that the value of R_{denom} does not necessarily have to be a constant in time. Rather, it merely has to be a constant across all users. The same value should be used for all users in a given slot when making a scheduling decision. The value would then be free to change for any future slots.

This leads back to the original question of what the value of the denominator should be. From before, we know that it must normalize the maxD metric to the range of the PF metric. It must now also be able to adapt to whatever the channel conditions might be. Note that the average rate value in the PF metric already accomplishes both of these goals. Thus, a logical choice for the denominator would be the mean value of the $\bar{R}_k(t)$ calculations. The updated equation for the maxD/PF metric would be as follows:

$$a\frac{DRC_k(t)}{\bar{R}_k(t)}+(1-a)\frac{DRC_k(t)}{\bar{R}_{av}(t)} \qquad (5.5)$$

The denominator $\bar{R}_{av}(t)$ is the mean of the $\bar{R}_k(t)$ average rate values across the set of users requesting service.

The following figure shows the average sector throughput per drop using the updated maxD/PF scheduling algorithm metric in the ITU pedestrian A channel.

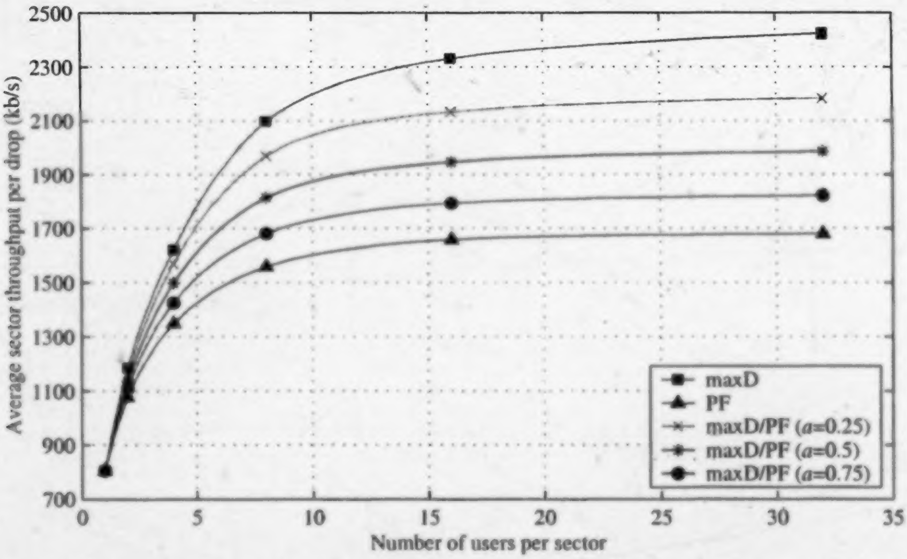


Figure 5.45: Average sector throughput per drop versus number of users per sector in ITU pedestrian A channel for maxD/PF scheduling algorithm with R_{denom} replaced by $\overline{R}_{av}(t)$

As can be seen by the above figure, the new denominator of $\overline{R}_{av}(t)$ appears to linearize the throughput range between the maximum rate and proportionally fair algorithms with the tuning parameter a quite well for all numbers of users. To confirm the effectiveness of the new denominator, the algorithm was tested in the ITU pedestrian B channel as well. The resulting throughputs are shown in the following figure.

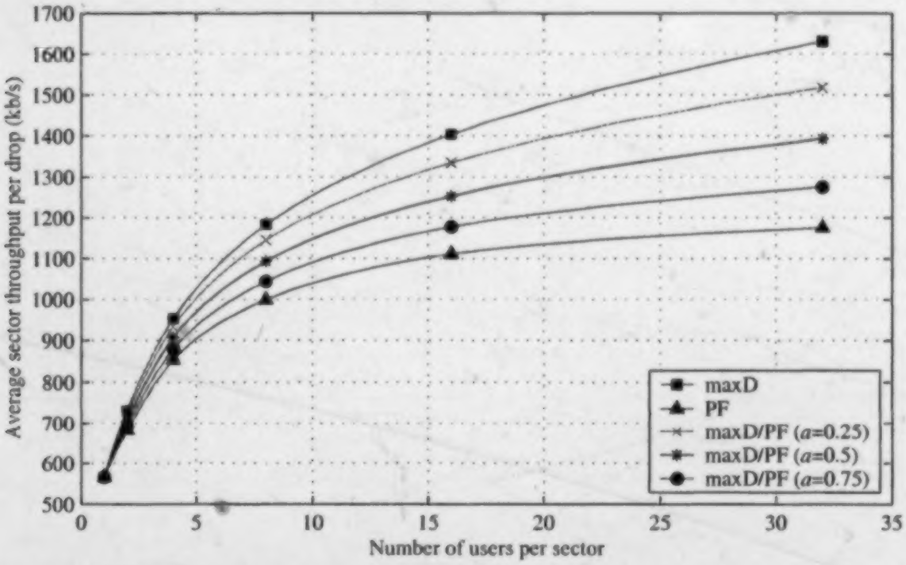


Figure 5.46: Average sector throughput per drop versus number of users per sector in ITU pedestrian B channel for maxD/PF scheduling algorithm with $\overline{R}_{av}(t)$ denominator

In the ITU pedestrian B channel, the three values used for the maxD/PF tuning parameter still effectively divided the throughput range into quarters. Therefore, the denominator of $\overline{R}_{av}(t)$ in the maxD/PF metric appears to have been an effective modification to the algorithm, as the resulting sector throughputs now match what was desired.

With the goal for the sector throughputs having been met, the remaining statistics for the algorithm can now be analyzed. The following two figures show the distributions of the average throughput per user per drop for the two pedestrian channels. As always, the case of 16 users per sector is assumed.

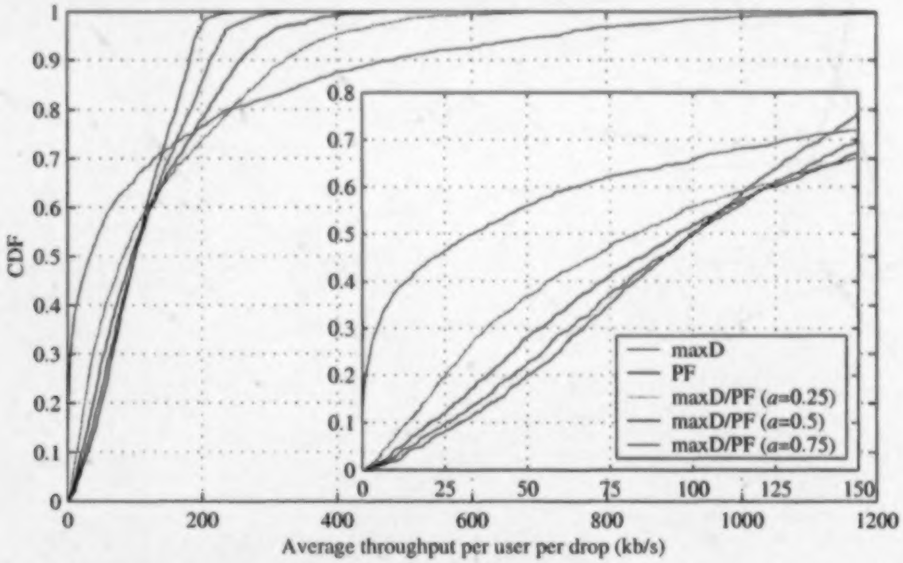


Figure 5.47: Distributions of average throughput per user per drop for the maxD/PF scheduling algorithm in the ITU pedestrian A channel with 16 users per sector

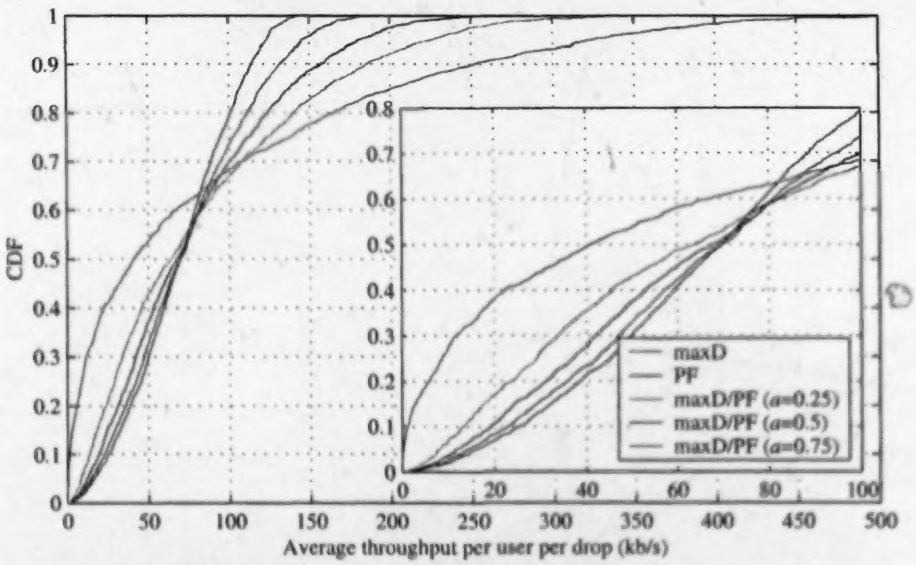


Figure 5.48: Distributions of average throughput per user per drop for the maxD/PF scheduling algorithm in the ITU pedestrian B channel with 16 users per sector

The following table gives the mean, median, and standard deviation of the above distributions.

TABLE XVI:
STATISTICS FOR AVERAGE THROUGHPUT PER USER PER DROP DISTRIBUTIONS FOR THE MAXIMUM RATE / PROPORTIONALLY FAIR HYBRID SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN A AND B CHANNELS WITH 16 USERS PER SECTOR

Channel	ITU pedestrian A					ITU pedestrian B				
Algorithm	maxD	maxD/ PF 0.25	maxD/ PF 0.5	maxD/ PF 0.75	PF	maxD	maxD/ PF 0.25	maxD/ PF 0.5	maxD/ PF 0.75	PF
Mean (kb/s)	145.7	133.3	121.7	112.1	103.6	87.7	83.4	78.3	73.6	69.5
Median (kb/s)	33.4	82.9	95.1	100.2	99.6	41.0	63.9	67.6	69.6	69.6
Std. Dev. (kb/s)	236.7	132.2	93.9	70.6	55.6	106.5	69.0	52.7	40.6	32.5

From the above distributions, it can be seen that there is a progression from the maximum rate distribution to the proportionally fair distribution as the tuning parameter *a* increases. However, that progression is no longer linear as it was in the case of the average sector throughputs. In general, as the tuning parameter increases, the mean throughput per user decreases, but the standard deviation also decreases, while the median increases. Hence, the distributions get tighter as more transmissions are allotted to the users in poorer channel conditions, bringing their throughputs up, although that increase comes at the expense of those users experiencing better channel conditions. Therefore, the tradeoff between throughput and fairness is seen once again.

The following graphs show the distributions of effective bit rates with 16 users per sector.

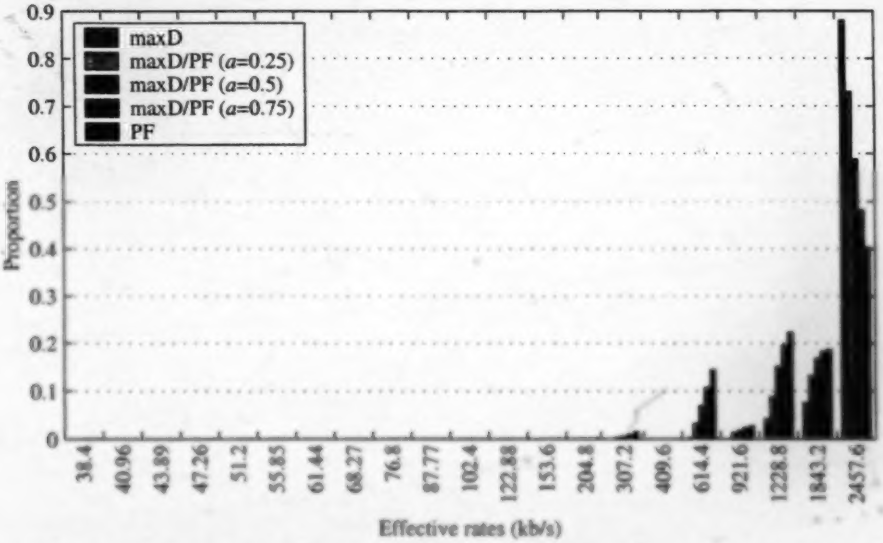


Figure 5.49: Distributions of effective bit rates for packets transmitted under the maxD/PF hybrid scheduling algorithms in the ITU pedestrian A channel with 16 users per sector

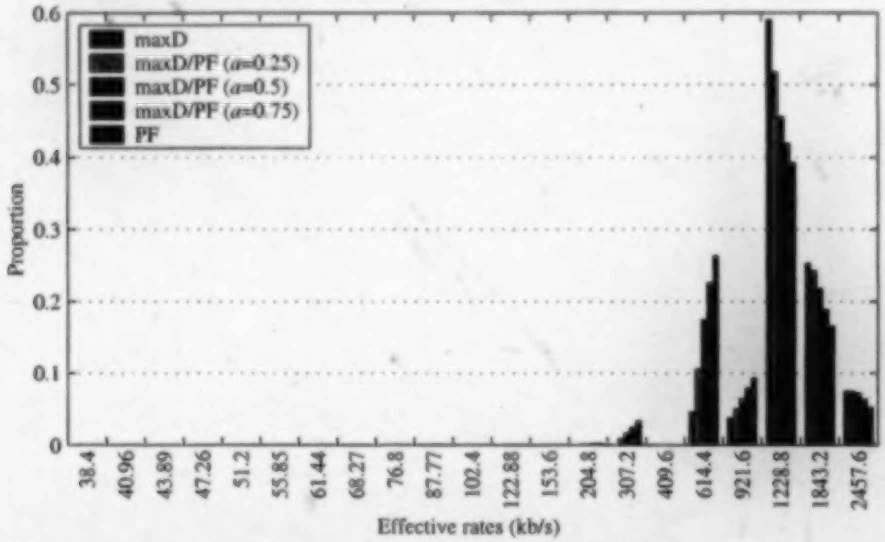


Figure 5.50: Distributions of effective bit rates for packets transmitted under the maxD/PF hybrid scheduling algorithms in the ITU pedestrian B channel with 16 users per sector

In conjunction with the effective bit rates, the following two graphs give the distributions of formats and number of transmissions used when sending packets, again with 16 users per sector.

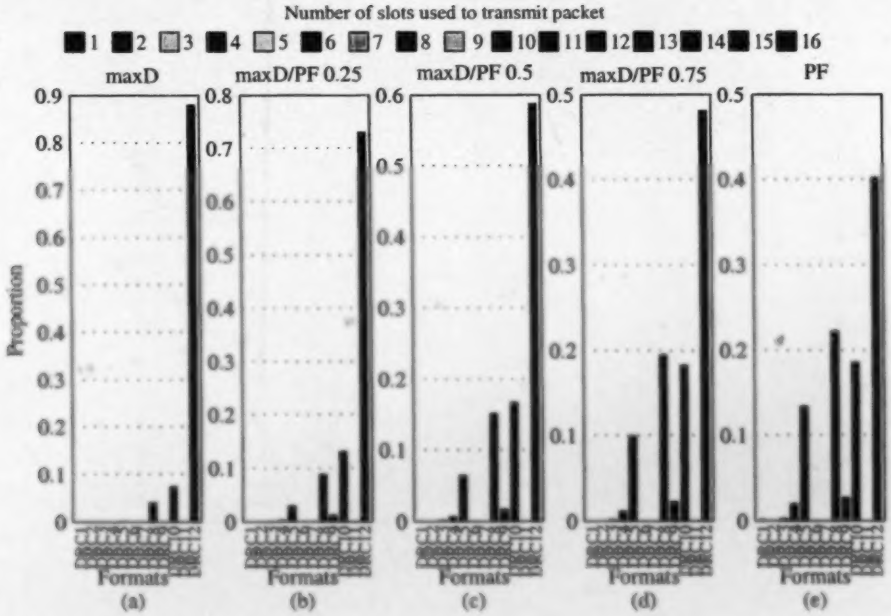


Figure 5.51: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a) maximum rate, (b),(c),(d) maxD/PF hybrid, and (e) proportionally fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector

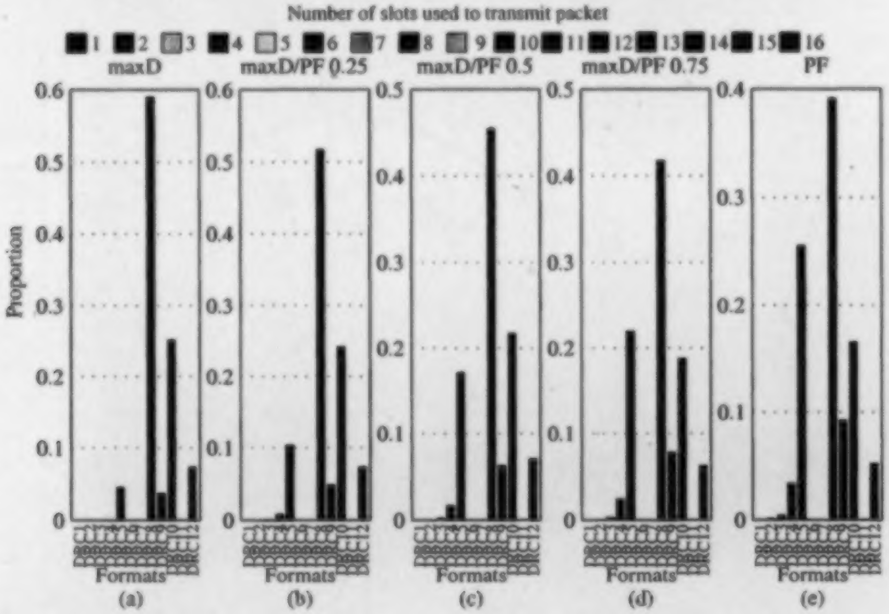


Figure 5.52: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a) maximum rate, (b),(c),(d) maxD/PF hybrid, and (e) proportionally fair scheduling algorithms in the ITU pedestrian B channel with 16 users per sector

As before, it can be seen that most of the transmissions occur at the higher bit rates, using the formats that allow only 1 or 2 slots maximum to send a packet.

In addition, there is an approximately linear progression between the maximum rate and the proportionally fair algorithms in the proportions of packets sent using each effective rate and each format. While this was not specifically a design goal for the algorithm, it is nonetheless a pleasing side effect. It is logical to expect this result as well, since if the average throughput per sector changes linearly, and each of the formats experiences approximately the same packet error rate, then on average, the distribution of the bit rates should be expected to change approximately linearly, as well.

The following two graphs show the average delay per packet per drop for the maxD/PF algorithm in the pedestrian channels.

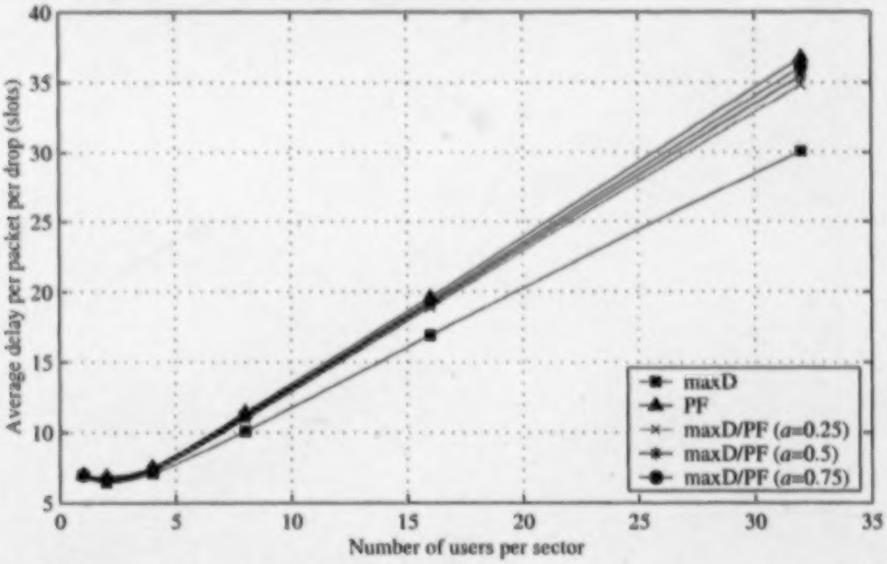


Figure 5.53: Average delay per packet per drop versus number of users per sector for the maxD/PF scheduling algorithm in the ITU pedestrian A channel

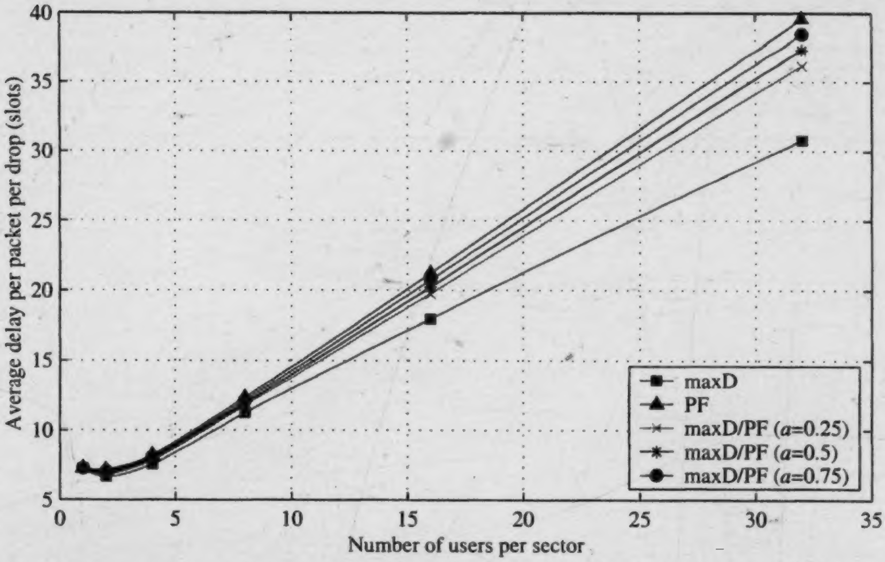


Figure 5.54: Average delay per packet per drop versus number of users per sector for the maxD/PF scheduling algorithm in the ITU pedestrian B channel

The average delay per packet appears to follow a more-or-less linear trend from the PF algorithm to the maxD/PF algorithm with $a = 0.25$. However, there is then a large jump for the maximum rate algorithm. This is likely because the maxD algorithm is an extremal case in terms of the service received by each user. Some users go without packets, while others receive multiple packets in rapid succession. Because of this large difference in service, the delay is much smaller than the other cases.

The two graphs below show the average packet delay per user per drop in comparison to the delay per packet shown above.

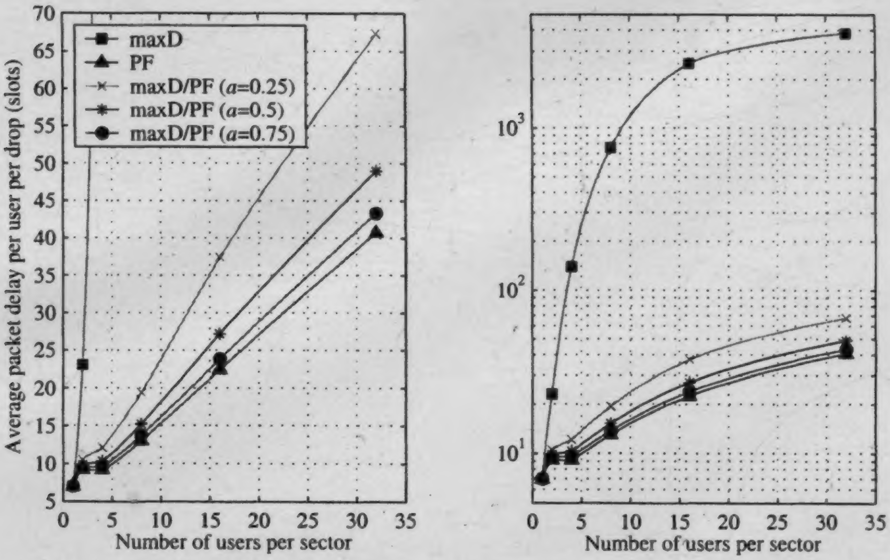


Figure 5.55: Average packet delay per user per drop versus number of users per sector for the maxD/PF scheduling algorithm in the ITU pedestrian A channel

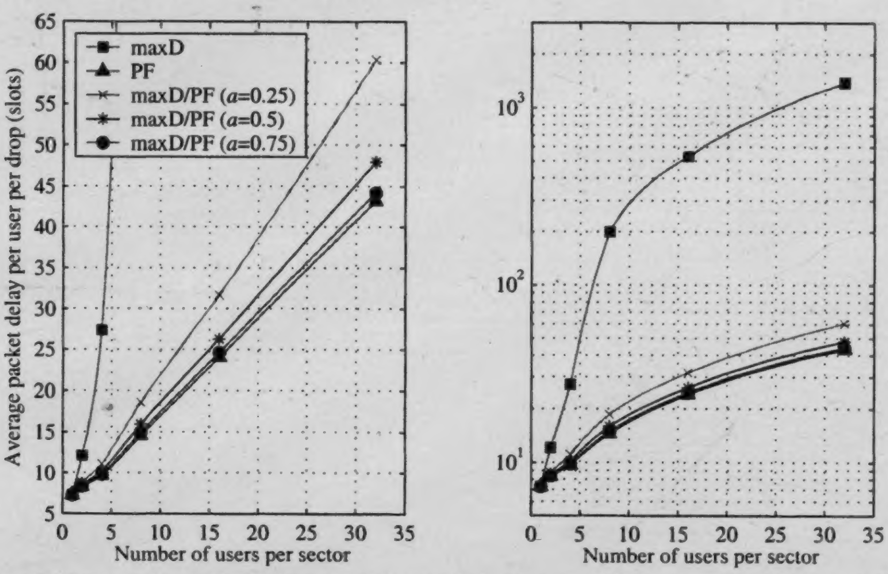


Figure 5.56: Average packet delay per user per drop versus number of users per sector for the maxD/PF scheduling algorithm in the ITU pedestrian B channel

The change in the delay per user is no longer linear with the tuning parameter, likely due to the differences in the number of packets received by each user. There is once again a large jump in the delay seen between the maximum rate algorithm and the other algorithms, although in this case the delay per user increases rather than decreasing in the case of the delay per packet. This again is due to the extreme unfairness of the maximum rate algorithm. This effect can also be explained by the metric for the algorithm. For any value of *a* other than 0, the proportionally fair half of the metric will come into play. For those users who have not yet received any data, the value of average rate in the denominator for those users will be extremely small. (The value is actually zero, but to avoid dividing by zero for the purposes of the simulation, those users with zero for an average rate instead used a value of 10^{-6} until they received their first packet.) Therefore, their overall value for the metric would be very large. This virtually guarantees that each user receives at least one packet. After the first packet is received, if the user receives no data for a sufficiently large window of time (i.e. on the order of t_c/a from the metric), that user would again start to have a high priority to receive a packet. This can quickly reduce the average delay seen per user with an increasing value of the tuning parameter. Hence, there is a sharp decrease in the delay between $a = 0$ and $a = 0.25$. For larger values of a, the maximum rate half of the algorithm becomes less significant, and the effective window size of the algorithm becomes diminishes at a smaller rate, and hence the delay changes much less as well.

The following diagrams show the distributions of delay per packet for the maxD/PF algorithm in the two pedestrian channels.

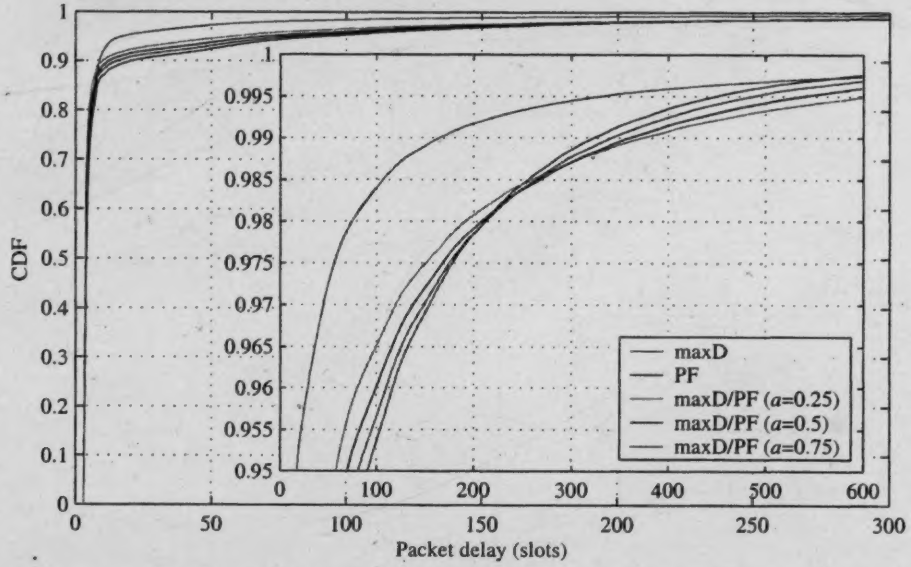


Figure 5.57: Cumulative distribution functions of delays experienced by packets under the maxD/PF scheduling algorithm in the ITU pedestrian A channel with 16 users per sector

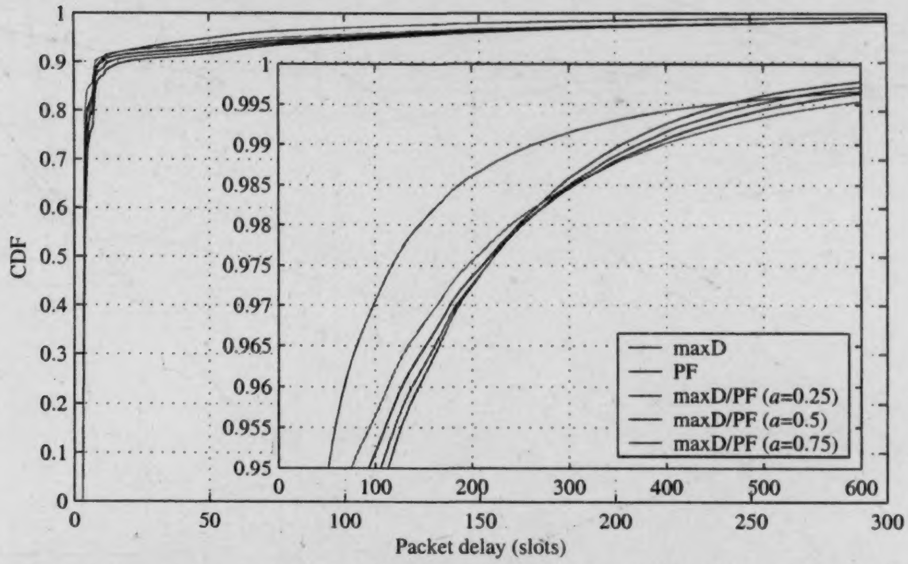


Figure 5.58: Cumulative distribution functions of delays experienced by packets under the maxD/PF scheduling algorithm in the ITU pedestrian B channel with 16 users per sector

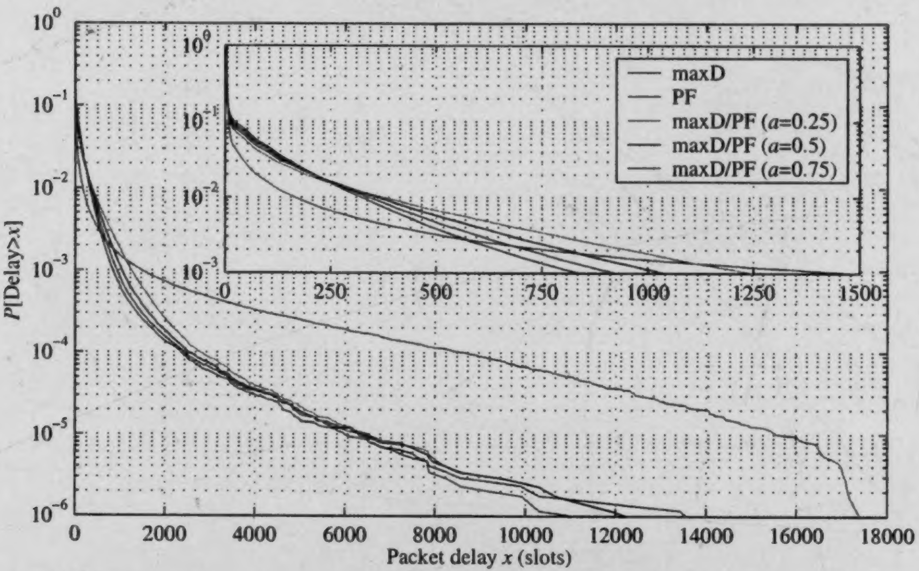


Figure 5.59: Probability of delays experienced by packets being larger than a given value under the maxD/PF scheduling algorithm in the ITU pedestrian A channel with 16 users per sector

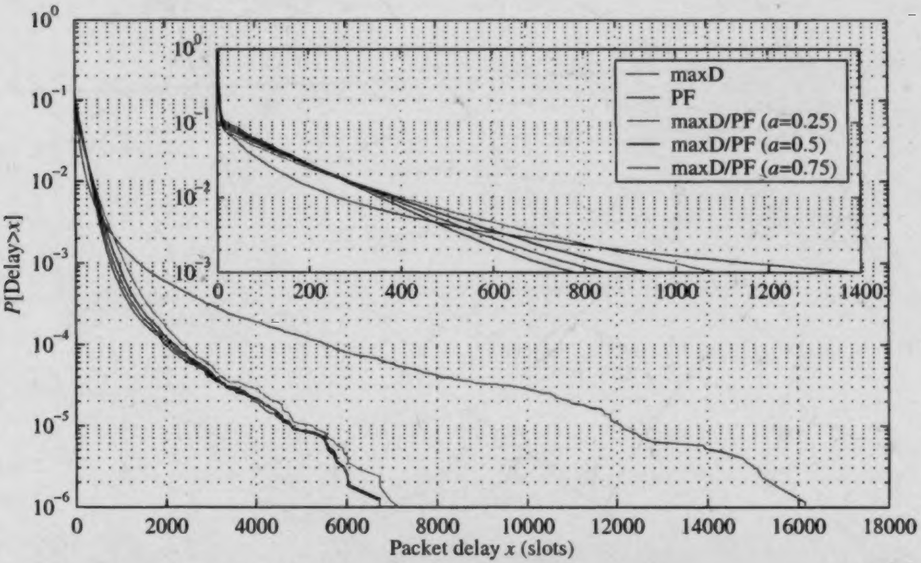


Figure 5.60: Probability of delays experienced by packets being larger than a given value under the maxD/PF scheduling algorithm in the ITU pedestrian B channel with 16 users per sector

A crossover can be seen in the distributions of packet delay with the varying tuning parameter. This crossover value occurs at a delay of about 250 slots in the ITU pedestrian A channel and about 275 slots in the ITU pedestrian B channel. For a given delay *x* below that crossover value, an increase in the tuning parameter corresponds to an increase in the probability that the delay will be larger than *x* (or alternately, a decreasing probability that the delay experience by a packet will be less than *x*). Beyond the crossover value, the reverse is true; increasing the tuning parameter will decrease the probability of experiencing a delay larger than *x*.

Note that the maxD algorithm does not conform to the same crossover value as for the other values of the tuning parameter, although it does still cross over. Overall, it has a much higher probability of low packet delays, while simultaneously displaying a much higher likelihood of extremely large delays.

The following table compares the percentiles and the standard deviations of the delays per packet.

TABLE XVII:
PERCENTILES AND STANDARD DEVIATIONS OF PACKET DELAYS FOR THE MAXD/PF SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN A AND B CHANNELS WITH 16 USERS PER SECTOR

Channel	ITU pedestrian A					ITU pedestrian B				
Algorithm / Percentile	maxD	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75	PF	maxD	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75	PF
50th (median)	4	4	4	4	4	4	4	4	4	4
90th	8.9	10.1	12.4	15.5	20.9	9.9	8.0	11.0	13.6	19.6
95th	17.3	57.9	69.5	80.5	90.4	51.1	74.8	92.8	105.2	113.0
99th	164.7	372.7	359.6	340.9	322.6	263.2	395.9	385.5	370.4	353.6
99.5th	329.3	606.6	537.3	485.0	448.3	449.6	576.7	527.8	491.9	461.3
99.9th	1476.4	1245.8	1037.0	922.2	837.9	1399.0	1085.6	937.6	843.0	777.2
99.95th	2835.4	1607.8	1338.0	1190.6	1095.7	2245.7	1362.2	1162.2	1069.3	998.5

99.99th	8673.6	2783.6	2516.8	2442.6	2345.1	5549.0	2348.8	2170.0	2087.2	1987.0
Std. Dev.	168.80	98.76	90.11	84.42	80.64	132.28	90.09	83.5	79.93	76.93

The above table once again reflects the crossover discussed earlier. For a given percentile *p*, the delay level at which *p%* of the transmitted packets experience that delay or less increases with an increasing tuning parameter, provided the percentile is small enough. From the graphs, the crossover percentile is at approximately 98.3 for both pedestrian channels. Above that value, increasing the tuning parameter causes a decrease in the delay at a constant percentile.

To compare with the distributions of delay per packet, the following two graphs give the distributions of delay per user with 16 users per sector.

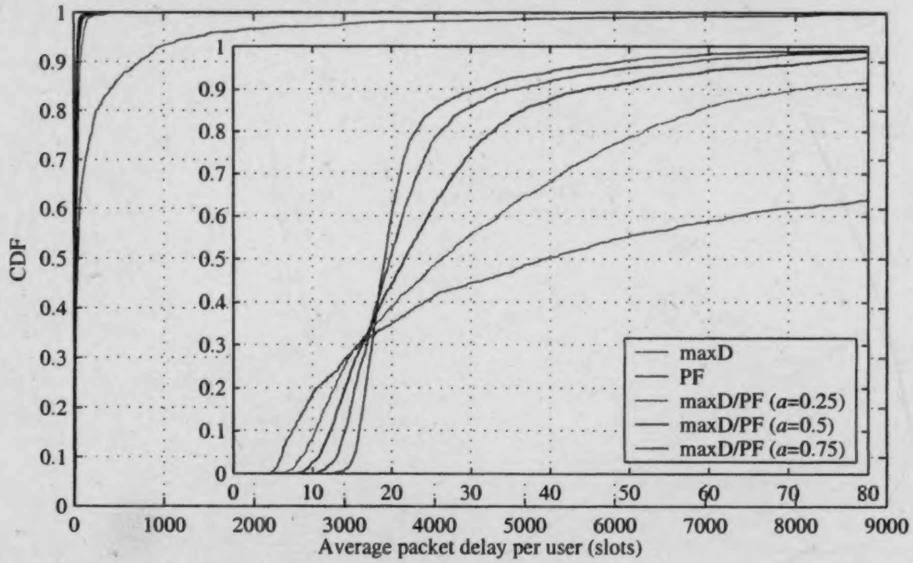


Figure 5.61: Distributions of average packet delay per user for maxD/PF scheduling algorithm in ITU pedestrian A channel with 16 users per sector

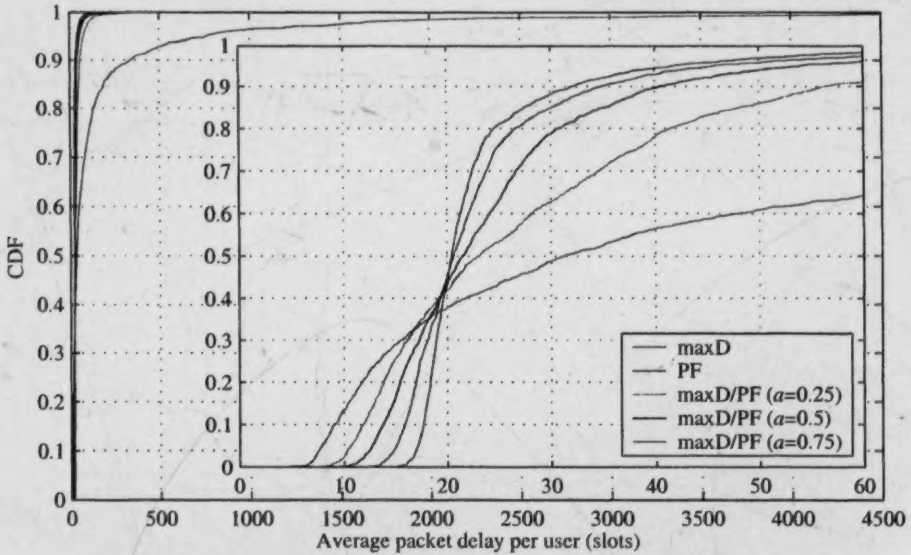


Figure 5.62: Distributions of average packet delay per user for maxD/PF scheduling algorithm in ITU pedestrian B channel with 16 users per sector

Once again we see a well-defined, although non-linear transition in the delay distributions with the changing tuning parameter. As the value of *a* increases, so does the tightness of the distribution. The extreme nature of the maximum rate algorithm is again revisited, as seen by the lengthy tail in the maxD distribution. The crossover characteristic can be seen again as well. In this case, however, the delay value of the crossover is much lower, as it the percentage at which it occurs (at about 17 to 19 slots and around 32% to 37%, respectively).

The following table numerically compares the distributions of the average packet delay per user.

TABLE XVIII:
PERCENTILES AND STANDARD DEVIATION OF AVERAGE PACKET DELAY PER USER FOR THE MAXD/PF SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN A AND B CHANNELS WITH 16 USERS PER SECTOR

Channel	ITU pedestrian A					ITU pedestrian B				
Algorithm Percentile	maxD	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75	PF	maxD	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75	PF
10th	7.9	10.4	12.3	14.1	15.9	9.1	11.8	13.8	15.9	17.7
25th	13.4	14.0	15.1	16.0	17.0	13.4	14.9	16.2	17.3	18.7
50th (median)	39.3	26.3	21.6	19.8	19.1	31.3	23.0	21.5	20.6	20.4
75th	186.4	46.1	29.9	24.5	21.8	102.7	38.2	28.3	24.7	23.2
90th	725.9	71.1	44.5	35.9	30.9	337.9	56.4	40.5	34.8	32.0
Std. Dev.	1102.1	40.7	24.3	17.8	14.1	710.5	32.9	25.5	23.2	21.9

The following two figures show the distributions of allocated slots per user per drop with 16 users per sector in the pedestrian channels.

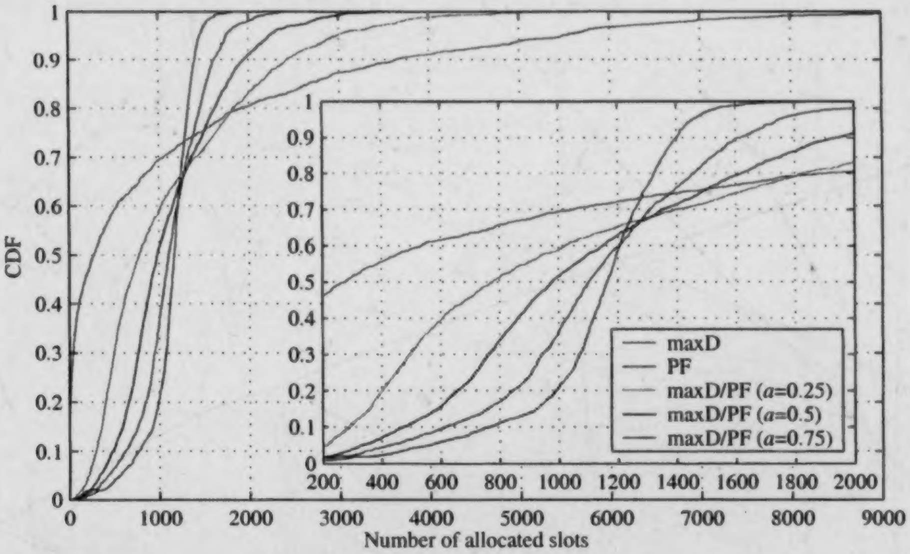


Figure 5.63: Distributions of allocated slots per user per drop for the maxD/PF scheduling algorithm in the ITU pedestrian A channel with 16 users per sector

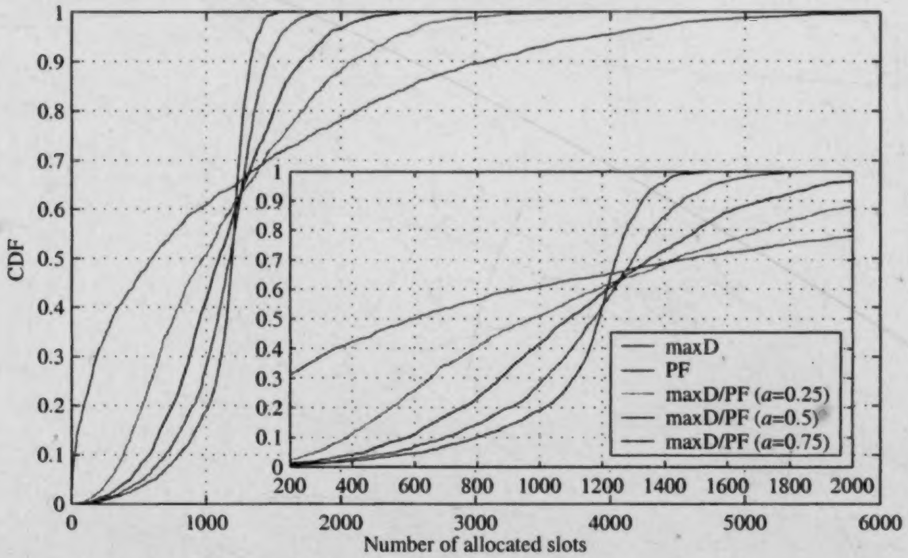


Figure 5.64: Distributions of allocated slots per user per drop for the maxD/PF scheduling algorithm in the ITU pedestrian B channel with 16 users per sector

The slot allocation distributions show the same trends as the other distributions seen earlier. Namely, there is a visible shift in the distributions as the tuning parameter changes, with the distributions becoming tighter as the value of a increases. The same crossover characteristic can be seen as well. For a quantitative comparison of the distributions, the following table is provided.

TABLE XIX:
PERCENTILES AND STANDARD DEVIATION OF SLOTS ALLOCATED PER USER PER DROP FOR THE MAXD/PF SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN A AND B CHANNELS WITH 16 USERS PER SECTOR

Channel	ITU pedestrian A					ITU pedestrian B				
Algorithm / Percentile	maxD	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75	PF	maxD	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75	PF
10th	0	289	473	617	766	21	386	578	698	797
25th	22	451	718	907	1040	140	590	818	971	1075
50th (median)	275	775	976	1101	1163	597	973	1093	1175	1190
75th	1433	1623	1487	1387	1290	1764	1574	1425	1326	1258
90th	3628	2419	1945	1615	1401	3085	2077	1715	1457	1328
Std. Dev.	1794.6	926.0	592.5	392.8	265.9	1280.3	678.2	444.1	303.5	231.9

Overall, the maxD/PF algorithm effectively performed its goal of linearizing the throughput range between the maximum rate and proportionally fair algorithms with the tuning parameter *a*. As a side effect, the distribution of effective bit rates also changed linearly with the tuning parameter. Other distributions, particularly those of delays, did not change linearly with the tuning parameter, although there was a distinct shift in the curves as the tuning parameter was adjusted. The average throughputs for sectors and users decreased with an increasing value for the tuning parameter, but so did the delays experienced by packets and users. The distributions of throughputs, delays, and slots also became tighter as the tuning parameter increased, indicating the algorithm was trading off decreased throughput for increased fairness amongst users.

5.3.3. Modified Largest Weighted Delay First (M-LWDF 1 & 2)

The two cases of the M-LWDF algorithm examined for this work essentially take the maximum rate metric (in the first case) or the proportionally fair metric (in the second case) and multiply by the head-of-line packet delay for each user. This user with the largest value of this product is selected for transmission. In the latter case, because the proportionally fair metric is involved, there exists the possibility of calculating the rate using either method described in the proportionally fair section. However, given that there was no significant difference between the two methods for the proportionally fair algorithm analysis, there is no reason to expect a significant difference with this algorithm either. Hence, only the "PF A" method was used in the analysis of the M-LWDF 2 algorithm.

The following two diagrams show the average throughput per sector for the M-LWDF algorithm in the pedestrian channels.

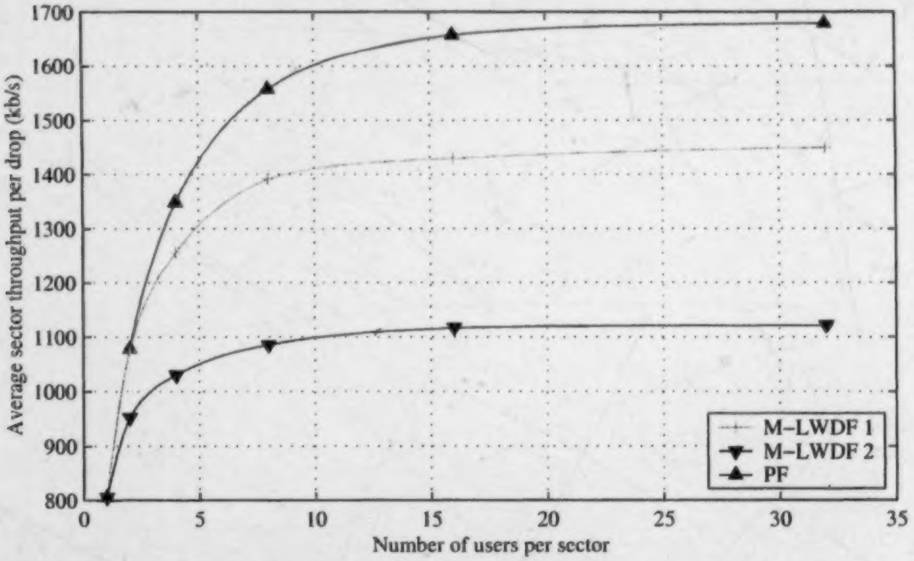


Figure 5.65: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian A channel for the M-LWDF 1 & 2 scheduling algorithms

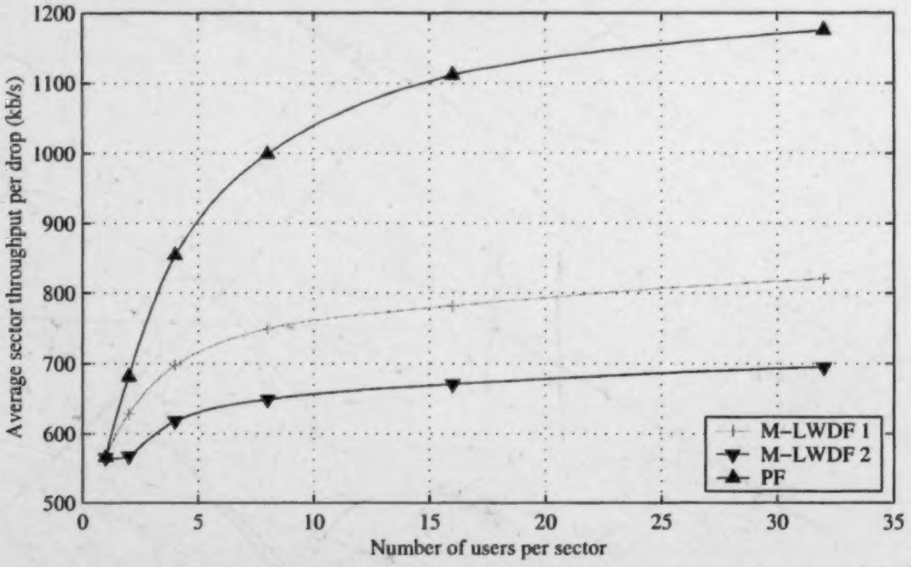


Figure 5.66: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian B channel for the M-LWDF 1 & 2 scheduling algorithms

As can be seen from the above diagrams, the throughput from the M-LWDF algorithm is significantly lower than that of the PF algorithm. This is no doubt due to the fact that the M-LWDF algorithm places a higher emphasis on delay. There is also much less multiuser gain to be seen. Between 1 and 32 users per sector, the throughput increases by a factor of 1.8 and 1.39 for the M-LWDF 1 and 2 algorithms respectively in the ITU pedestrian A channel, and a factor of

1.45 and 1.23 respectively in the ITU pedestrian B channel. The rate of multiuser gain with increasing users also flattens out much faster than for the PF algorithm.

The following graphs show the distributions of throughput per user per drop for the M-LWDF algorithm with 16 users per sector in the pedestrian channels.

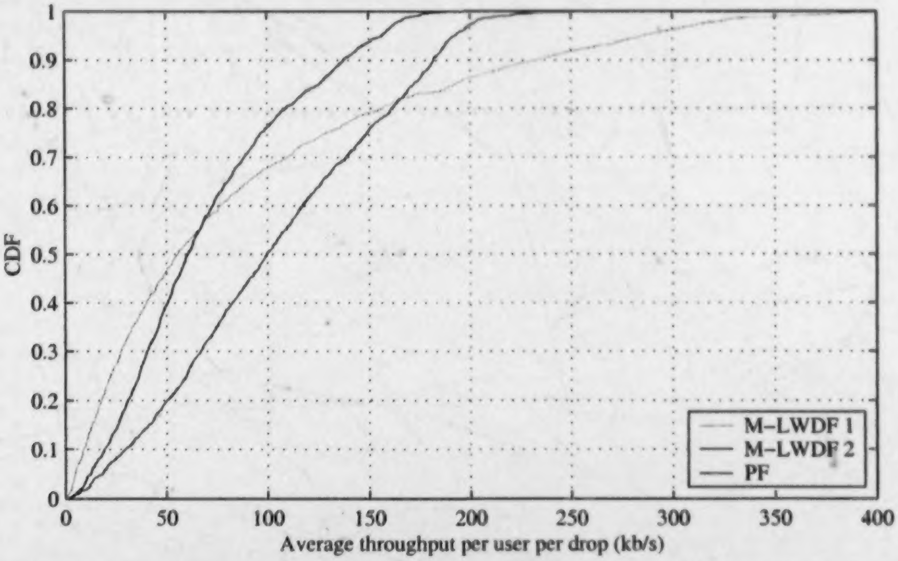


Figure 5.67: Distributions of average throughput per user per drop for the M-LWDF scheduling algorithm in the ITU pedestrian A channel with 16 users per sector

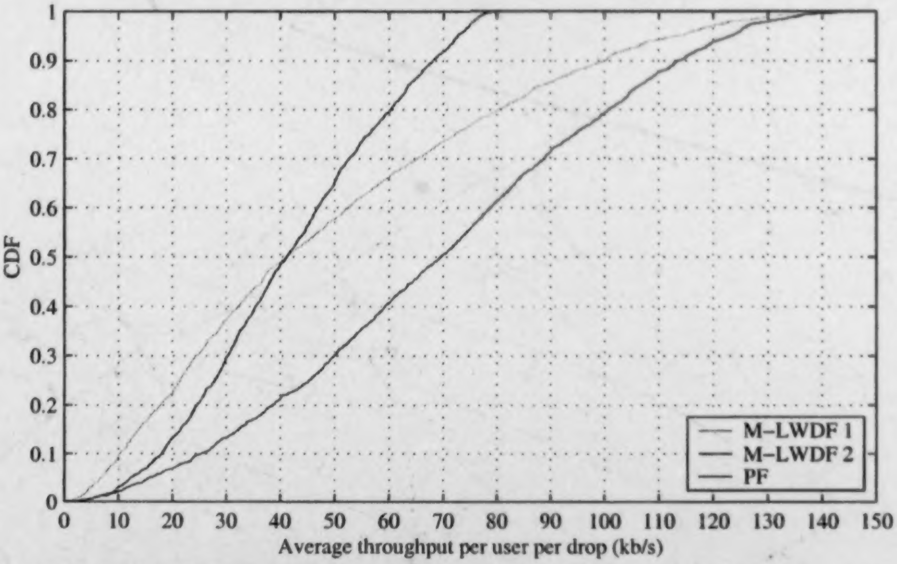


Figure 5.68: Distributions of average throughput per user per drop for the M-LWDF scheduling algorithm in the ITU pedestrian B channel with 16 users per sector

The M-LWDF 1 distribution is somewhat spread out, although nowhere near as extensively as the maxD algorithm was. The M-LWDF 2 algorithm actually has a rather tight distribution of throughputs, even more so than the PF algorithm. However, both cases of the M-LWDF algorithm show a comparatively high probability of relatively lower throughputs than the PF algorithm. These points can be further illustrated by examining the mean, median, and standard deviation of the user throughputs.

TABLE XX:
STATISTICS FOR AVERAGE THROUGHPUT PER USER PER DROP DISTRIBUTIONS FOR THE M-LWDF 1 & 2 SCHEDULING ALGORITHMS IN THE ITU PEDESTRIAN A AND B CHANNELS WITH 16 USERS PER SECTOR

Channel	ITU pedestrian A			ITU pedestrian B		
Algorithm	M-LWDF 1	M-LWDF 2	PF	M-LWDF 1	M-LWDF 2	PF
Mean (kb/s)	89.3	69.8	103.6	48.8	41.9	69.5
Median (kb/s)	55.4	60.5	99.6	41.4	41.2	69.6
Std. Dev. (kb/s)	88.9	43.5	55.6	33.6	18.4	32.5

Numerically, both the mean and the median throughput values for the two M-LWDF cases are lower than for the PF algorithm. The standard deviation is higher for the M-LWDF 1 algorithm than either of the other two cases, once again giving a feel for the spread of the values, although the excess spread is not as substantial in the ITU pedestrian B channel when compared to the PF algorithm. In both channels, the standard deviation of the M-LWDF 2 algorithm is substantially smaller, giving a further indication of how tight the distribution is.

The following figures show the distributions of the effective bit rates for the M-LWDF algorithms with 16 users per sector.

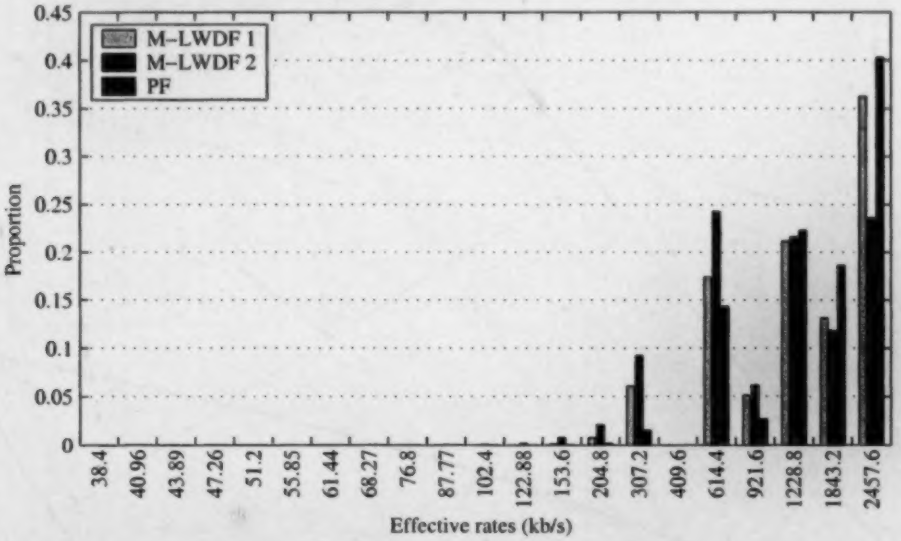


Figure 5.69: Distributions of effective bit rates for packets transmitted under the M-LWDF scheduling algorithms in the ITU pedestrian A channel with 16 users per sector

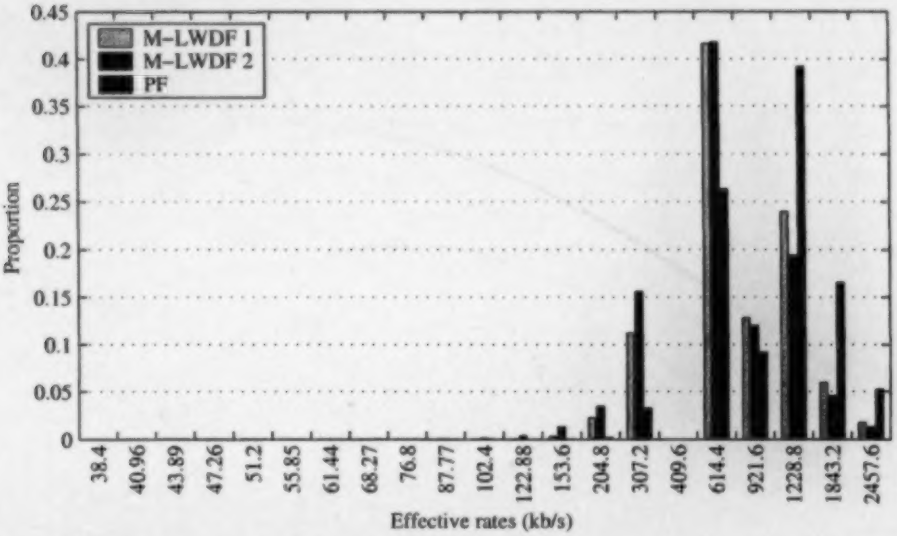


Figure 5.70: Distributions of effective bit rates for packets transmitted under the M-LWDF scheduling algorithms in the ITU pedestrian B channel with 16 users per sector

The next two figures show the distributions of formats and number of transmissions for correctly received packets under the M-LWDF scheduling algorithms.

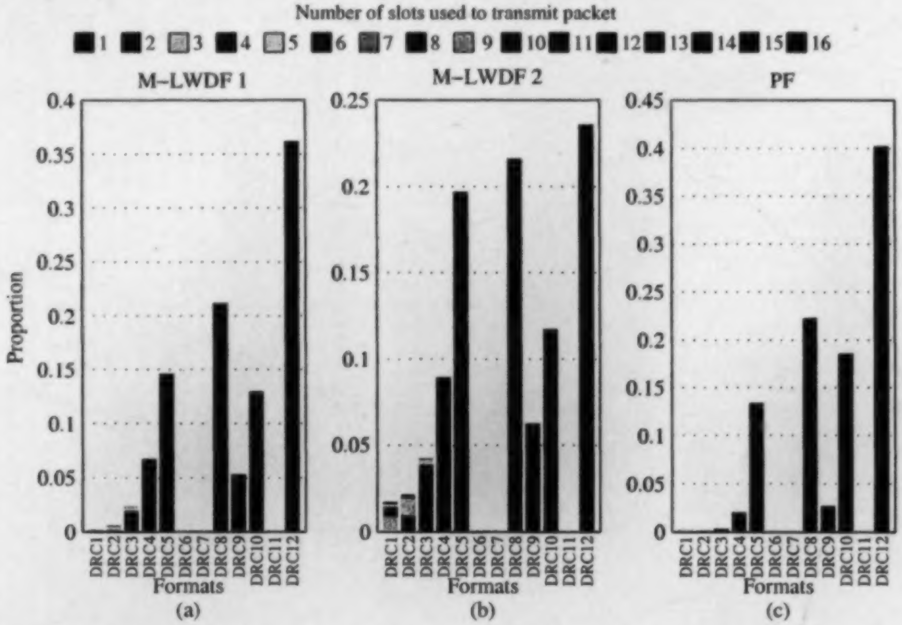


Figure 5.71: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a) M-LWDF 1, (b) M-LWDF 2, and (c) proportionally fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector

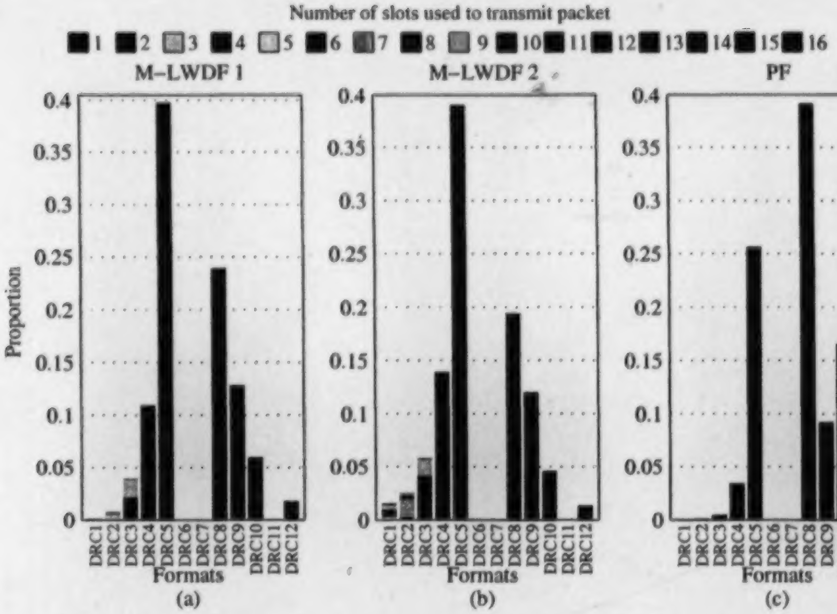


Figure 5.72: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a) M-LWDF 1, (b) M-LWDF 2, and (c) proportionally fair scheduling algorithms in the ITU pedestrian B channel with 16 users per sector

As earlier indicated with the sector throughputs, the effective bit rates for the M-LWDF algorithms are much lower than for the PF algorithm. The formats selected are also those using lower rates, especially in the case of the M-LWDF 2 algorithm. This is further evidence of how the delay portion of the M-LWDF algorithm overrides the channel conditions and selects users requesting lower rates if their packet delay is sufficiently high.

The next two figures show the average delay per packet per drop as a function of the number of users per sector for the M-LWDF algorithms.

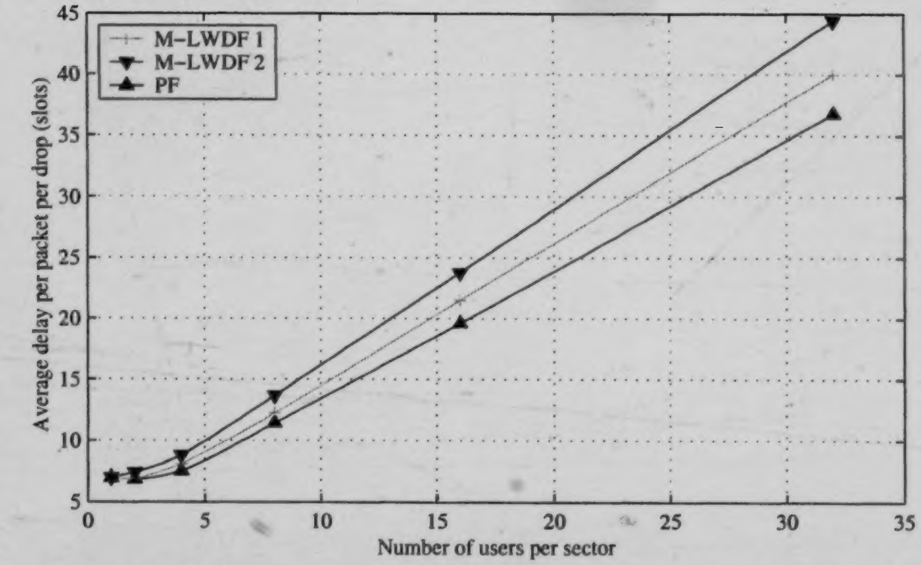


Figure 5.73: Average delay per packet per drop versus number of users per sector for the M-LWDF scheduling algorithms in the ITU pedestrian A channel

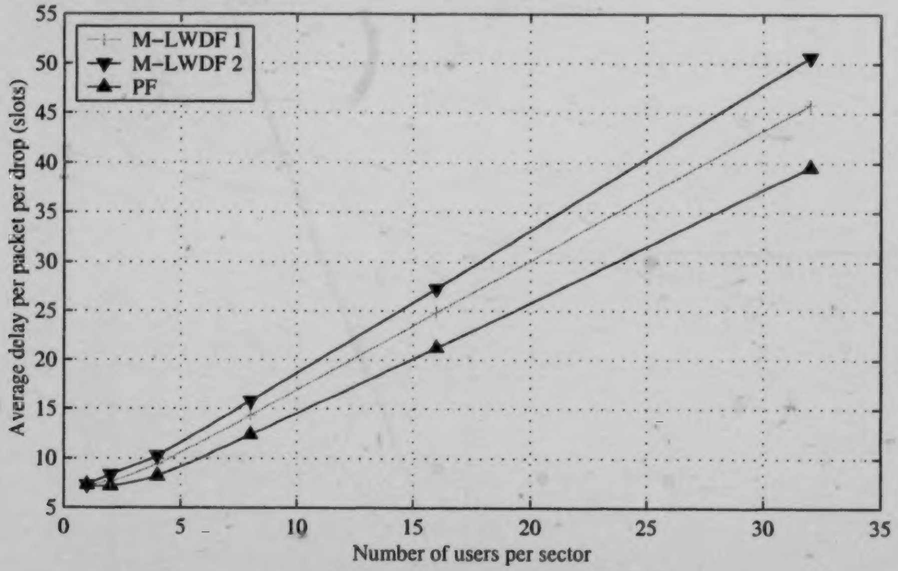


Figure 5.74: Average delay per packet per drop versus number of users per sector for the M-LWDF scheduling algorithms in the ITU pedestrian B channel

The relationship between higher/lower throughput and lower/higher average packet delay is repeated for the M-LWDF algorithms. Since the throughput for the M-LWDF algorithms is substantially lower than that for the PF algorithm, the average delay per packet can be expected to be (and is) larger under the M-LWDF algorithms than under the PF algorithm.

The following two diagrams give the average packet delay per user for the M-LWDF algorithms.

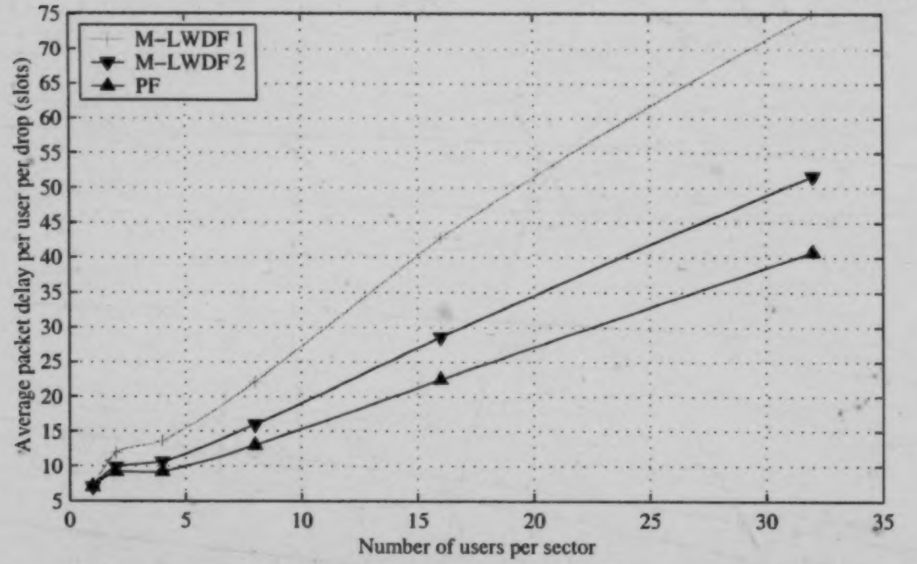


Figure 5.75: Average packet delay per user per drop versus number of users per sector for the M-LWDF scheduling algorithms in the ITU pedestrian A channel

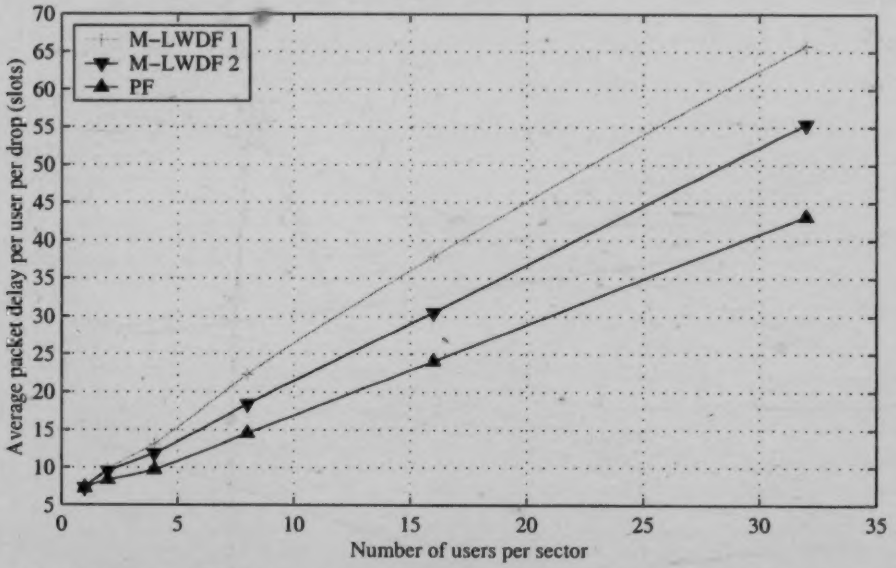


Figure 5.76: Average packet delay per user per drop versus number of users per sector for the M-LWDF scheduling algorithms in the ITU pedestrian B channel

For the delays per user, we see that the values for the M-LWDF algorithm are substantially greater than for the PF algorithm. This is the opposite effect of the previously observed trend. In general, as the throughput decreased, so did the delay per user. There was one previous exception to that trend, however; the delay per user for the round robin algorithm was quite high. The reasons for the longer delays are also similar. Any time that an algorithm very often overrides the channel conditions due to some other factor (i.e. delay) taking precedence, there is the likelihood that the algorithm will do so at an inopportune time for that packet. The more often the override occurs, the more likely this is to happen, and as a consequence, the average delay a user experiences will increase. This is the primary reason for the increased delay under the M-LWDF 2 algorithm. It also plays a role in the M-LWDF 1 algorithm, but the main reason why the delay per user is the highest for the M-LWDF 1 algorithm is because it is a modification of the maxD algorithm. Therefore, considering the users with the worst channel conditions, until the delay for a packet is large enough, the corresponding user will not receive any service. When the delay is sufficiently large, only one packet for that user will be transmitted. The next packet for that user will not be sent until the delay builds up again. Therefore, most of the users in the system still experience fairly lengthy delays, and hence the average delay per user is quite high. The delay is much improved when compared to the maxD algorithm, but that improvement comes at the expense of the throughput.

The following graphs show the distributions of average delay per packet per drop with 16 users per sector under the M-LWDF algorithms.

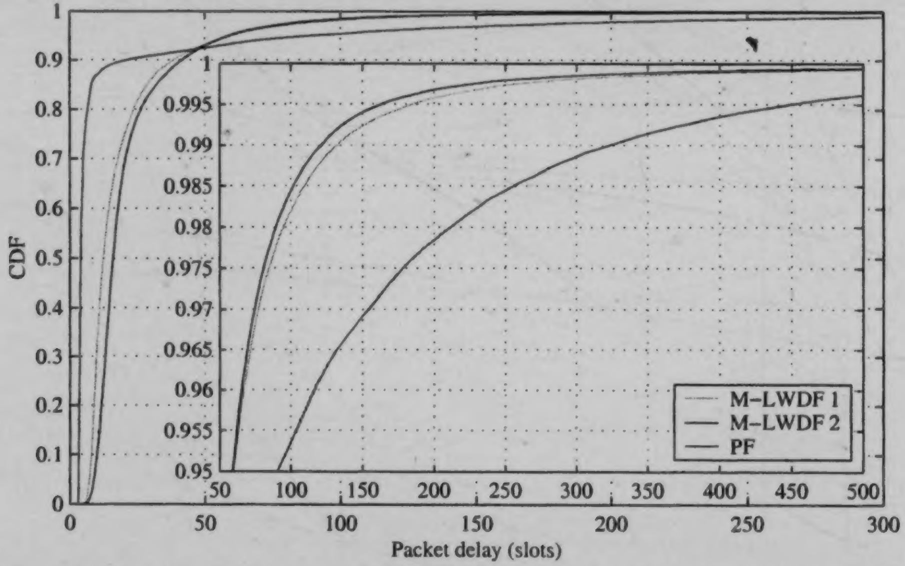


Figure 5.77: Cumulative distribution functions of delays experienced by packets under the M-LWDF scheduling algorithms in the ITU pedestrian A channel with 16 users per sector

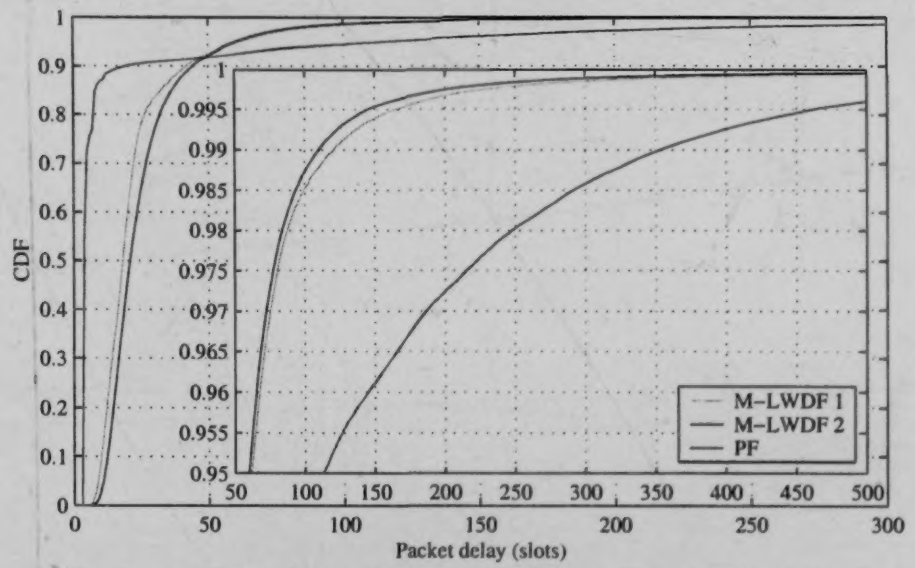


Figure 5.78: Cumulative distribution functions of delays experienced by packets under the M-LWDF scheduling algorithms in the ITU pedestrian B channel with 16 users per sector

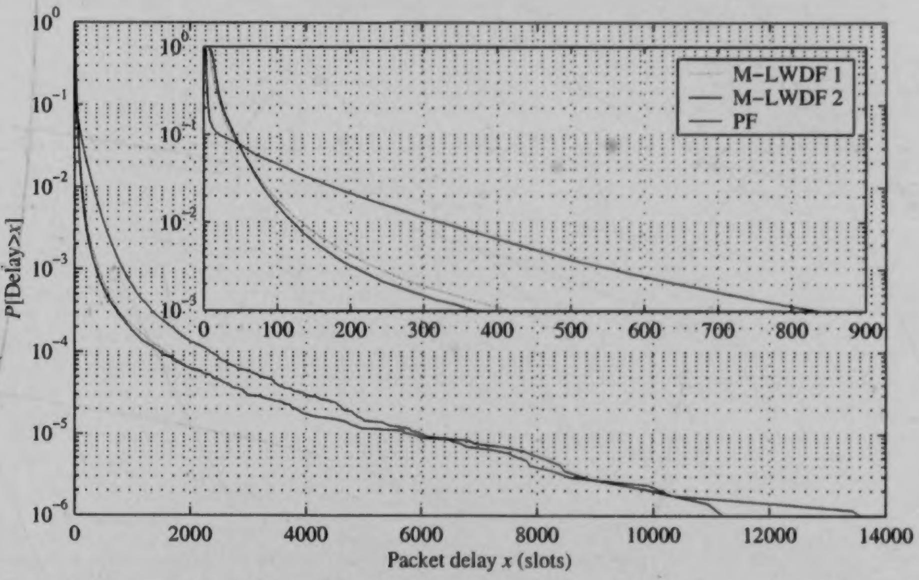


Figure 5.79: Probability of delays experienced by packets being larger than a given value under the M-LWDF scheduling algorithms in the ITU pedestrian A channel with 16 users per sector

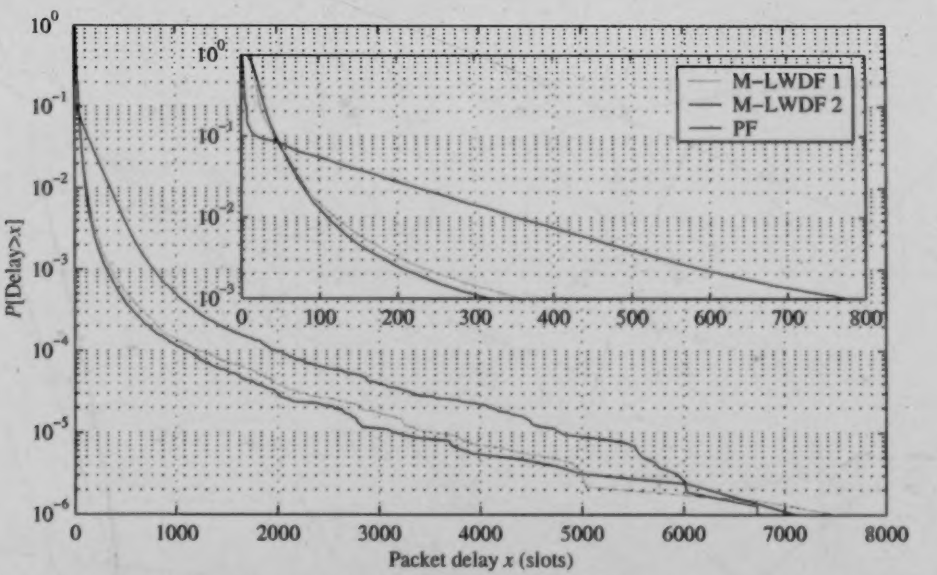


Figure 5.80: Probability of delays experienced by packets being larger than a given value under the M-LWDF scheduling algorithms in the ITU pedestrian B channel with 16 users per sector

The above graphs show the main strength of the M-LWDF algorithm. Although the average delay is somewhat high when compared to the PF algorithm, the probability of a packet experiencing an extremely large delay under the M-LWDF algorithm is much lower. Beyond a delay of about 50 slots, the probability of experiencing a delay equal to or greater than that delay will be smaller under either of the M-LWDF algorithms than under the PF algorithm. There is still a much better chance of a packet experiencing a delay less than 50 slots under the PF algorithm than under the M-LWDF algorithms, however. The following table gives the percentiles and standard deviations of the above distributions.

TABLE XXI:
PERCENTILES AND STANDARD DEVIATIONS OF PACKET DELAYS FOR THE M-LWDF SCHEDULING ALGORITHMS IN THE ITU PEDESTRIAN A AND B CHANNELS WITH 16 USERS PER SECTOR

Channel	ITU pedestrian A			ITU pedestrian B		
Algorithm / Percentile	M-LWDF 1	M-LWDF 2	PF	M-LWDF 1	M-LWDF 2	PF
50th (median)	12.3	15.8	4	18.2	20.9	4
90th	38.6	41.5	20.9	41.9	44.3	19.6
95th	59.6	59.2	90.4	61.5	59.9	113.0
99th	132.3	121.2	322.6	120.5	110.2	353.6
99.5th	184.8	163.0	448.3	165.0	145.0	461.3
99.9th	419.6	373.0	837.9	361.5	317.0	777.2
99.95th	596.1	549.8	1095.7	521.5	447.7	998.5
99.99th	1579.1	1479.7	2345.1	1223.4	1103.2	1987.0
Std. Dev.	48.53	47.54	80.64	37.78	34.15	76.93

The fact that the median packet delay is not 4 slots for the M-LWDF 1 & 2 algorithms is indicative of how the delay consideration can override the channel conditions. In most of the algorithms seen previously, the median delay of 4 slots indicated that most of the time, multiple packets were sent to the same user in succession. With the median above 4 slots for the M-LWDF algorithms, this shows that the grouping of packets is being broken up by transmissions to other users. The delay factor would be the cause of this. Also note that for the higher percentiles, the value of the percentile delay for the M-LWDF algorithms is around half or slightly above half of that for the PF algorithm. For a given delay, the probability of the delay being larger than that value can be up to an order lower in magnitude for the M-LWDF algorithms when compared to the PF algorithm. The smaller standard deviations also show that the M-LWDF distributions are tighter and that there are fewer variations in the delays experienced by all packets.

The following two graphs show the distributions of average packet delay per user per drop with 16 users per sector in the pedestrian channels under the M-LWDF scheduling algorithms.

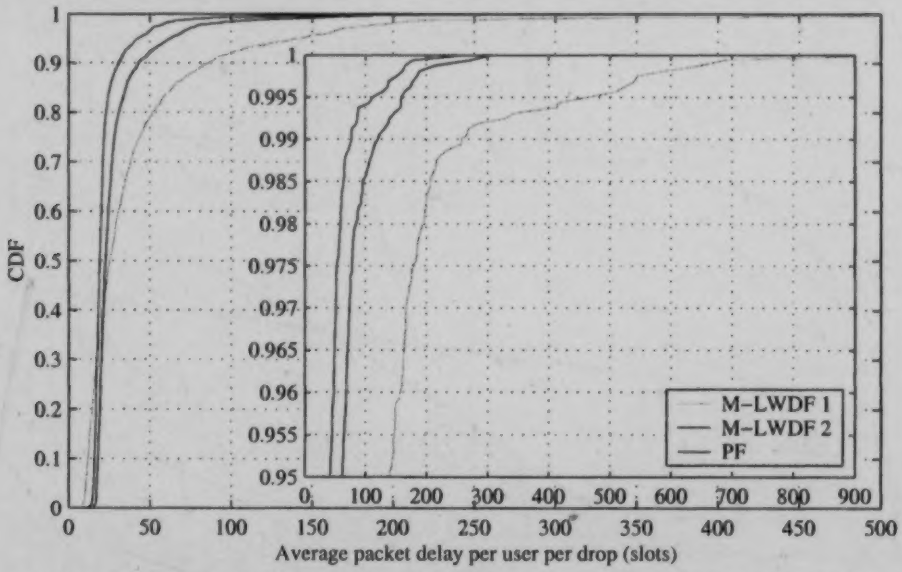


Figure 5.81: Distributions of average packet delay per user for M-LWDF scheduling algorithms in ITU pedestrian A channel with 16 users per sector

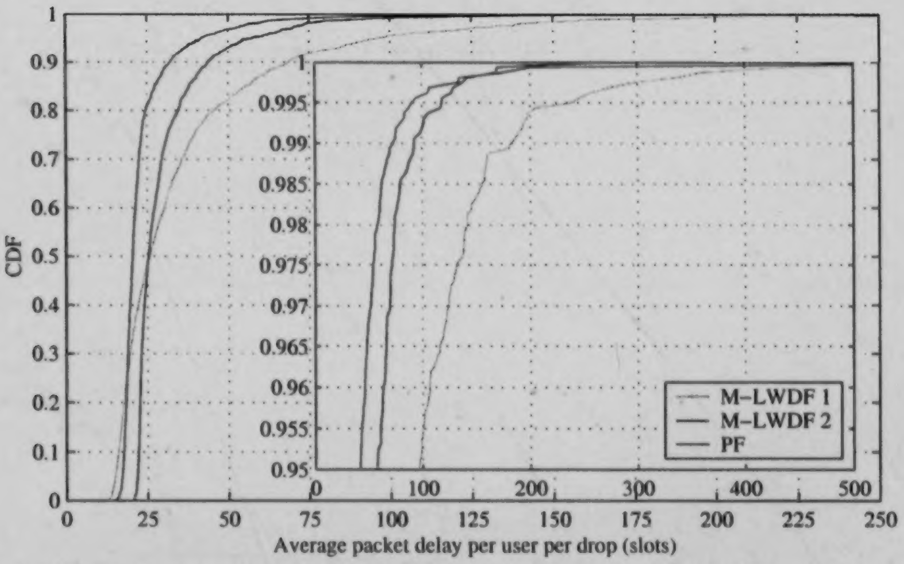


Figure 5.82: Distributions of average packet delay per user for M-LWDF scheduling algorithms in ITU pedestrian B channel with 16 users per sector

As earlier indicated by the average user delays for various numbers of users per sector, the distributions also show that the average packet delay experienced by each user is larger under the M-LWDF algorithms than under the PF algorithm. This is particularly true for the M-LWDF 1 algorithm. The following table gives the percentiles and standard deviations of delays per user.

TABLE XXII:
PERCENTILES AND STANDARD DEVIATION OF AVERAGE PACKET DELAY PER USER FOR THE M-LWDF SCHEDULING ALGORITHMS IN THE ITU PEDESTRIAN A AND B CHANNELS WITH 16 USERS PER SECTOR

Channel	ITU pedestrian A			ITU pedestrian B		
Algorithm / Percentile	M-LWDF 1	M-LWDF 2	PF	M-LWDF 1	M-LWDF 2	PF
10th	12.0	18.0	15.9	16.0	22.1	17.7
25th	15.9	19.6	17.0	18.5	22.7	18.7
50th (median)	25.3	22.7	19.1	25.7	25.4	20.4
75th	43.7	28.1	21.8	39.5	31.3	23.2
90th	86.2	42.9	30.9	68.8	43.7	32.0
Std. Dev.	61.1	20.7	14.1	59.9	15.5	21.9

Other than at the lower percentiles for the M-LWDF 1 algorithm, the M-LWDF algorithms consistently show delays per user larger than those for the PF algorithm. There is a much larger standard deviation for the M-LWDF 1 algorithm, likely as a result of its being a modification of the maxD algorithm. Interestingly, the standard deviation for the M-LWDF 2 algorithm is smaller than that of the PF algorithm in the ITU pedestrian B channel. Because of the larger fluctuations in the ITU pedestrian B channel, there is correspondingly a larger spread in the average packet delay per user values for the proportionally fair algorithm across all users. This results in a tail at the upper end of the distribution, as some users experience larger delays. The delay factor in the M-LWDF 2 algorithm removes this tail, resulting in a tighter distribution and hence a smaller standard deviation.

The two graphs below show the distributions of slots per user per drop for the M-LWDF algorithms with 16 users per sector.

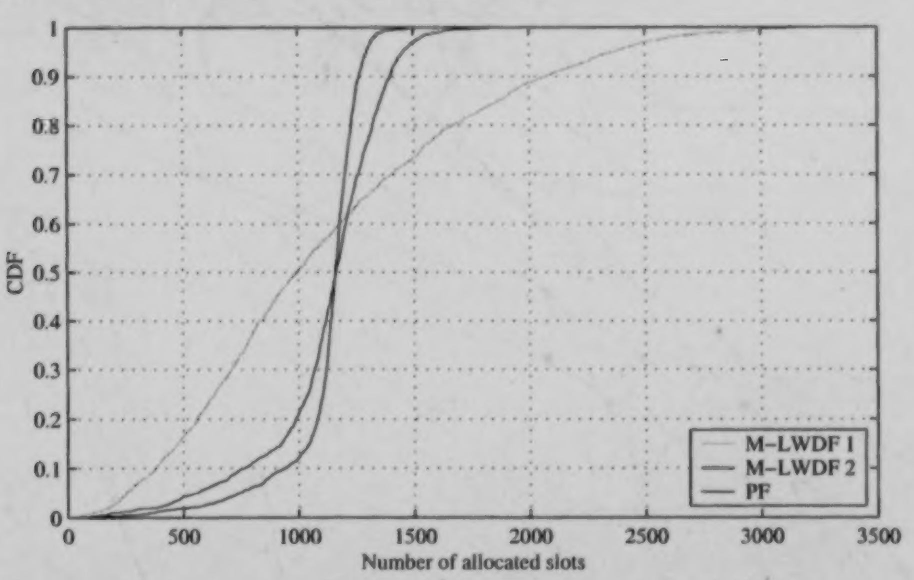


Figure 5.83: Distributions of allocated slots per user per drop for the M-LWDF scheduling algorithms in the ITU pedestrian A channel with 16 users per sector

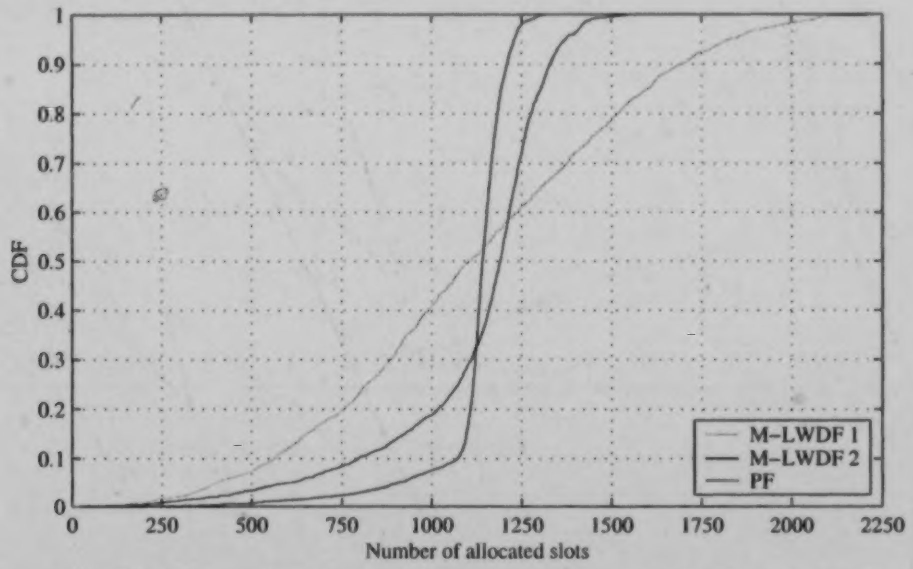


Figure 5.84: Distributions of allocated slots per user per drop for the M-LWDF scheduling algorithms in the ITU pedestrian B channel with 16 users per sector

The following table gives the percentiles and the standard deviations for the above distributions.

TABLE XXIII:
PERCENTILES AND STANDARD DEVIATION OF SLOTS ALLOCATED PER USER PER DROP FOR THE M-LWDF SCHEDULING ALGORITHMS IN THE ITU PEDESTRIAN A AND B CHANNELS WITH 16 USERS PER SECTOR

Channel	ITU pedestrian A			ITU pedestrian B		
Algorithm / Percentile	M-LWDF 1	M-LWDF 2	PF	M-LWDF 1	M-LWDF 2	PF
10th	383	942	766	565	1072	797
25th	636	1104	1040	820	1116	1075
50th (median)	999	1161	1163	1103	1141	1190
75th	1533	1217	1290	1443	1173	1258
90th	2072	1266	1401	1691	1210	1328
Std. Dev.	646.4	175.3	265.9	428.8	114.1	231.9

Once again, there is a wide spread of values observed for the M-LWDF 1 algorithm. Given the throughput and delay per user distribution spreads seen earlier, along with the maxD basis for the algorithm, this was to be expected. However, the tightness of the M-LWDF 2 algorithm is of particular interest. The M-LWDF 2 metric is the product of the normalized requested rate and the head-of-line queuing delay. Recall that for the M-LWDF 2 algorithm, a large proportion of the packets were transmitted using the lower rate formats. For these formats, the rate times the maximum number of slots required to transmit the packet is a constant. Therefore, it is not much of a stretch to expect that the rate times the queuing delay would be fairly constant as well. The tightness of this distribution has been hinted at earlier, as the standard deviations of the other M-LWDF 2 distributions have been consistently lower that the corresponding PF standard deviations. This is an overall indication that the M-LWDF 2 algorithm is an extremely fair algorithm.

Overall, the two M-LWDF algorithms place a high priority on delay, as it is part of the metric used to make scheduling decisions. As a result, there is a substantial decrease in throughput when compared to the PF algorithm. The average delay per packet and average packet delay per user also increase. This occurs because the channel conditions are being overridden by the delay component, and as a result, packets are transmitted at suboptimal times. However, the probability of any given packet experiencing a very long delay is substantially lowered. In the case of the M-LWDF 2 algorithm, the fairness of the algorithm is also improved. The standard deviation in the distributions of throughputs, delays per packet and per user, and slots is lower than for the PF algorithm. Whether the improved fairness and decreased probability of long delays is worth the tradeoff in the throughput and average delays would depend on the situation that the algorithm is used in. If a user was receiving a real-time data stream, such as a voice connection (e.g. videophone, voice over IP, etc.), the need for small delays might warrant the other tradeoffs.

5.3.4. Slot-wise Fair ("slot-fair") and Packet-wise Fair ("packet-fair")

The goal of the slot-wise fair and packet-wise fair algorithms is to maximize the sector throughput under the constraint that each user requesting service should receive approximately the same proportion of slots/packets. This is achieved by having the scheduler choose in slot *t* the user *k* for transmission with the highest value of the sum $U_k(t)+v_k(t)$. The $U_k(t)$ term is the requested rate of user *k* in slot *t*, normalized to 38.4 kb/s. This term is considered by the authors of the algorithm in [20] to be the "value" or how worthwhile it is to transmit to user *k* in slot *t*. The term $v_k(t)$ is an adaptive parameter that keeps track of the proportion of slots/packets each user is allocated.

There is a parameter *a* within $v_k(t)$ to control how quickly the algorithm adapts to changes in the proportions of slots/packets that each user receives. In order to determine a proper value for that parameter, a few quick trials were run with different tuning parameters. These tests were run using 16 users per sector and perfect multislot prediction in order to avoid having to iterate the simulations to obtain suitable SIR margins. Giving a cursory look at the resulting throughputs and slot/packet distributions, it was determined that a=0.25 was a good choice for the slot-fair algorithm, while a=0.5 was better for the packet-fair algorithm.

Some initial results for the average sector throughput of the slot-fair and packet-fair algorithms are shown in the two figures below.

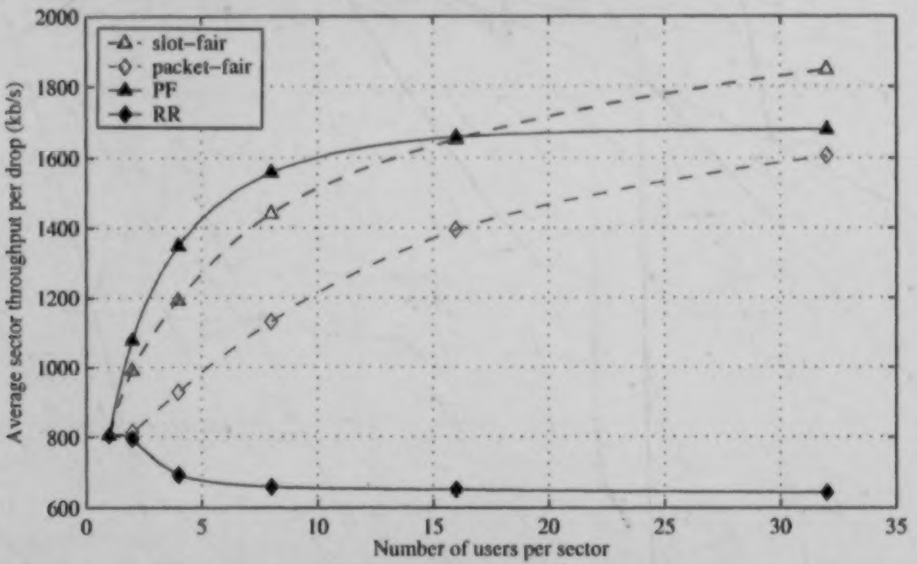


Figure 5.85: Initial average sector throughput per drop vs. number of users per sector results in the ITU pedestrian A channel for the slot-fair and packet-fair scheduling algorithms

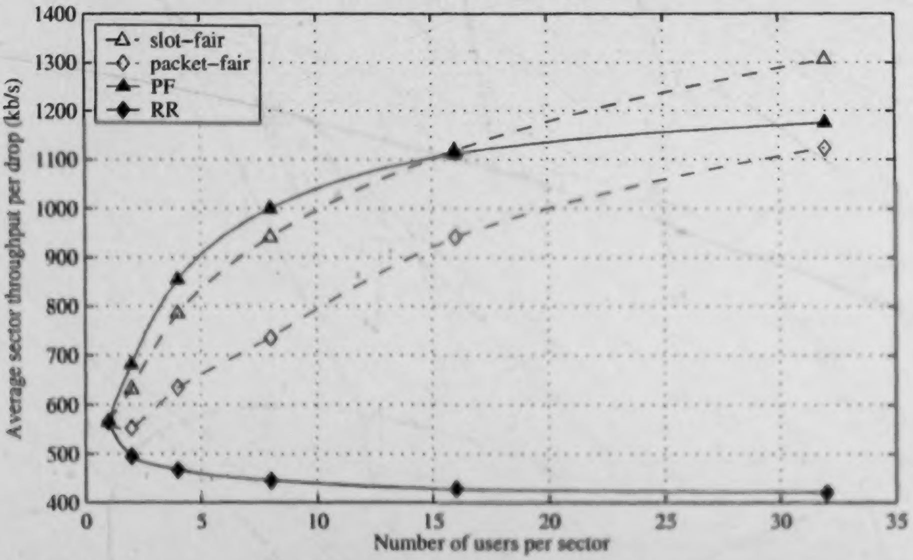


Figure 5.86: Initial average sector throughput per drop vs. number of users per sector results in the ITU pedestrian B channel for the slot-fair and packet-fair scheduling algorithms

Notice the interesting dip in throughput for the packet-fair algorithm at 2 users per sector. This point was characterized by a larger than expected packet error rate. More precisely, the average packet error rate per user was approximately double what is normally acquired at 2 users per sector, even though the average packet error rate per format still remained around 1% for all formats. This effect was initially thought to be a quirk in the operation of the algorithm. The cause for this increased error rate was not discovered until the simulations using perfect prediction of the channel were run. With perfect prediction, there should be a very small packet error rate (i.e. well below 1%), as the users know exactly what rate they can support. However, the average packet error rate was still unusually high at 2 users per sector, even with the perfect prediction.

The cause of the higher packet error rates was traced to how the algorithm handles users that go into outage for extended periods of time. During that time, their $U_k(t)$ value is 0. However, the $v_k(t)$ value for that user continuously increases during that period; the user receives no service, so its proportion of the allocated packets/slots decreases, thereby increasing the value of $v_k(t)$ so as to give that user higher priority to receive a transmission. Given a long enough period of time, the sum of $0+v_k(t)$ for that user can become larger than the other user(s) who can support a non-zero rate. Therefore, despite the fact that the user will almost definitely not be able to receive the packet correctly, a transmission is still scheduled to that user using the lowest possible rate. These transmissions to an outage user are what causes the higher average packet error rates per user. Although overall 1% of the packets sent with the lowest rate format are in error, certain users wind up with an extremely large packet error rate. This skews the average packet error rate upwards. The side effect is that the sector throughput drops. Interestingly, the delay seen by packets and by users also drops. By scheduling those packets to outage users, they are removed from the head of the queue, allowing a new packet in its place. Therefore, despite the fact that the packet is not received correctly, the overall delay seen on average is smaller than if the system waited until the packet could be received correctly.

The effects seen here are as a result of the metric for the algorithm being a sum rather than a product. As a result, even a user with a zero requested rate might be selected for transmission. Note that this effect is not seen with larger numbers of users per sector. In that event, if a user goes into outage for an extended period of time, it is far more likely that there is another user that can support a high enough rate that that user's sum will be larger than the outage user's sum. It is therefore highly unlikely that the outage user will be selected for transmission before it returns from its outage state and can support a non-zero rate again.

In order to correct the algorithm, it should be made so that no packets are sent to users who are in outage (unless all users in the sector are in outage). This can be done rather simply by changing the $U_k(t)$ term for an outage user. Rather than having the term equal zero, it can instead equal a large negative value (e.g. -9,999,999 as used in the simulations). This ensures that the sum of $U_k(t)+v_k(t)$ for an outage user will never be larger than a non-outage user, and hence the outage user will not be selected for transmission. In the event that all users are in outage, the user who received the smallest proportion of packets at that point in time would be the user selected to receive data (at the lowest possible rate), since that user would have the largest value of $v_k(t)$, and hence the largest sum.

The following two graphs show the average sector throughput using the corrected algorithms.

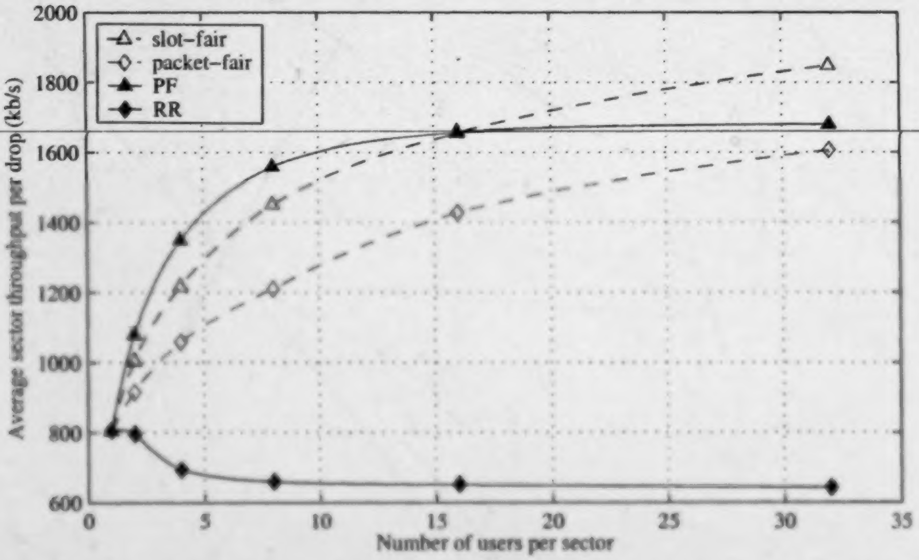


Figure 5.87: Average sector throughput per drop vs. number of users per sector results in the ITU pedestrian A channel for the corrected slot-fair and packet-fair scheduling algorithms

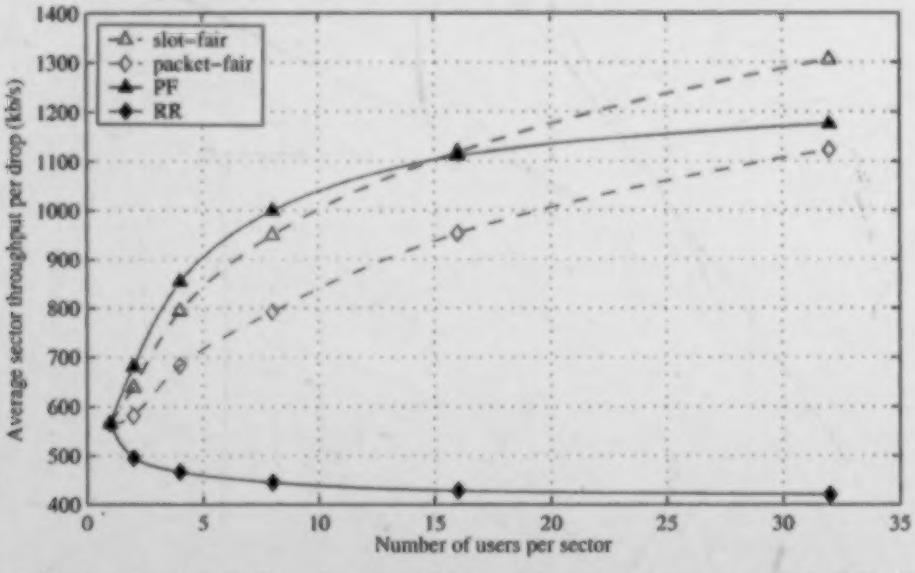


Figure 5.88: Average sector throughput per drop vs. number of users per sector results in the ITU pedestrian B channel for the corrected slot-fair and packet-fair scheduling algorithms

Notice that the bump at 2 users per sector is no longer present for the packet-fair algorithm. Instead, the throughput curve rises much more smoothly across all numbers of users. There is also a minor increase in the throughput for the slot-fair algorithm at 2 and 4 users per sector.

The throughput achieved by the slot-fair algorithm is comparable to that achieved by the PF algorithm for the two pedestrian channels. Below 16 users per sector, the slot-fair throughput is less that that for the PF algorithm, while above

16 users per sector, the slot-fair throughput is greater. The crossover in throughput occurs at about 16 users per sector (i.e. the throughputs are approximately the same at that point). This would seem to indicate that below 16 users per sector, the PF algorithm gains somewhat in throughput through a more unequal allocation of slots compared to the slot-fair algorithm. Above 16 users per sector, the requirement of choosing the user with the highest instantaneous-to-average ratio of signal strength results in a suboptimal allocation of resources. As the slot-fair algorithm indicates, it is possible to allocate slots more fairly and still yield a larger sector throughput.

Considering the packet-wise fair algorithm, the average sector throughput is much improved when compared to the round robin algorithm. At the same time, both algorithms attempt to provide the same number of packets to all users requesting service. In fact, the packet-fair algorithm accomplishes this goal better than the RR algorithm, as the packet-fair algorithm is able to compensate those users who go into outage and fall behind in the proportion of packets they receive. Therefore, the packet-fair algorithm is an improvement over the round-robin algorithm both in terms of throughput and in terms of fairness.

The next two diagrams show the distributions of throughput per user for the slot-fair and packet-fair algorithms with 16 users per sector.

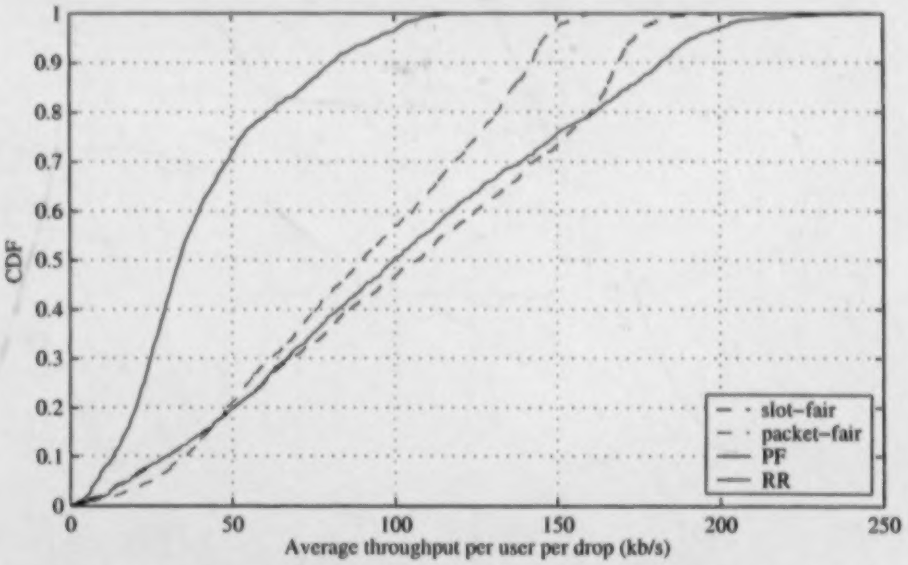


Figure 5.89: Distributions of average throughput per user per drop for the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector

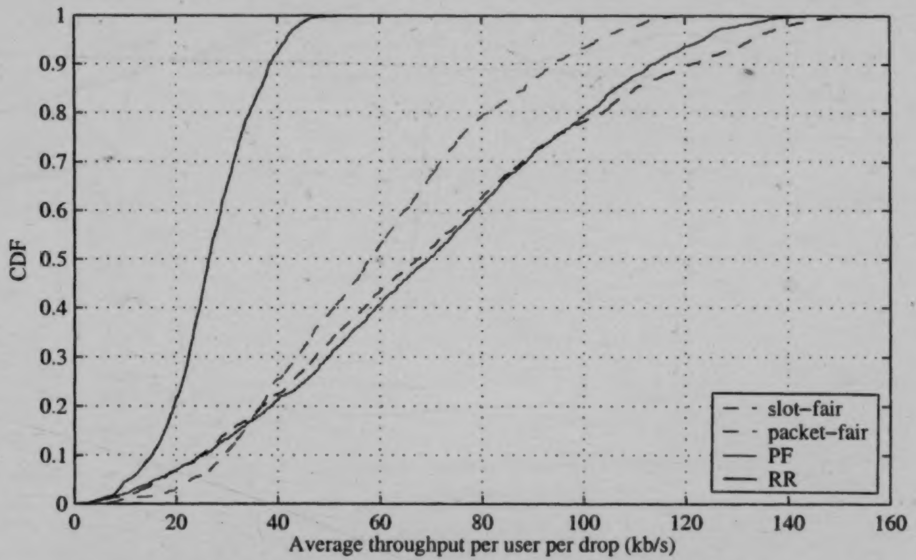


Figure 5.90: Distributions of average throughput per user per drop for the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian B channel with 16 users per sector

The distribution of the average throughput per user is very much the same for the slot-fair algorithm as for the PF algorithm. Since the average throughput per sector is very close for both algorithms, at 16 users per sector, this result is not all that unexpected. Of more interest is the distribution for the packet-fair algorithm. The throughputs achieved by the users under that algorithm are much higher than under the round robin algorithm. This is another demonstration of how taking channel conditions into account will markedly improve the throughput of a system. The following table gives some statistics of the above throughput distributions.

TABLE XXIV:
STATISTICS FOR AVERAGE THROUGHPUT PER USER PER DROP DISTRIBUTIONS FOR THE SLOT-WISE FAIR AND PACKET-WISE FAIR SCHEDULING ALGORITHMS IN THE ITU PEDESTRIAN A AND B CHANNELS WITH 16 USERS PER SECTOR

Channel	ITU pedestrian A				ITU pedestrian B			
Algorithm	slot-fair	packet-fair	PF	RR	slot-fair	packet-fair	PF	RR
Mean (kb/s)	103.4	89.2	103.6	40.7	69.9	59.6	69.5	26.7
Median (kb/s)	106.9	90.1	99.6	33.7	67.7	58.2	69.6	26.5
Std. Dev. (kb/s)	52.0	40.1	55.6	25.6	35.1	24.4	32.5	8.95

The following figures show the distributions of effective bit rates for the slot-fair and packet-fair algorithms with 16 users per sector.

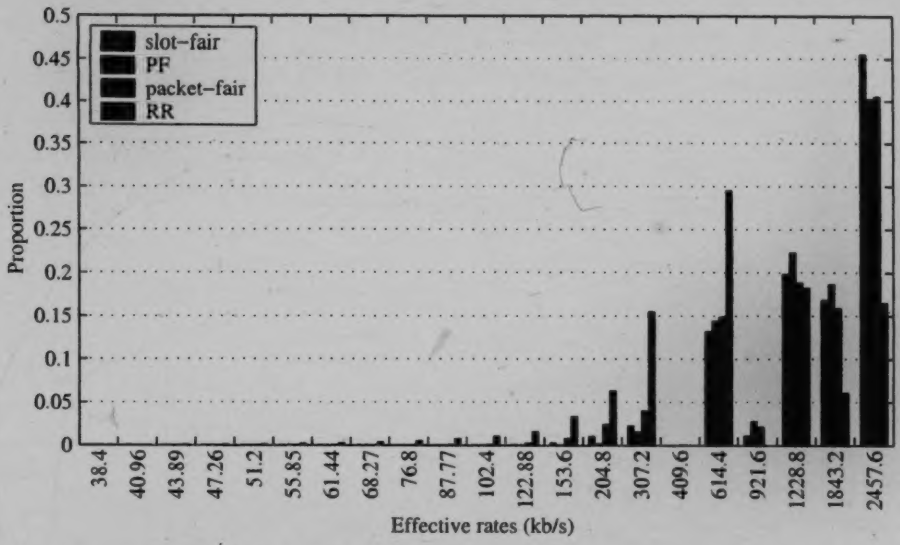


Figure 5.91: Distributions of effective bit rates for packets transmitted under the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector

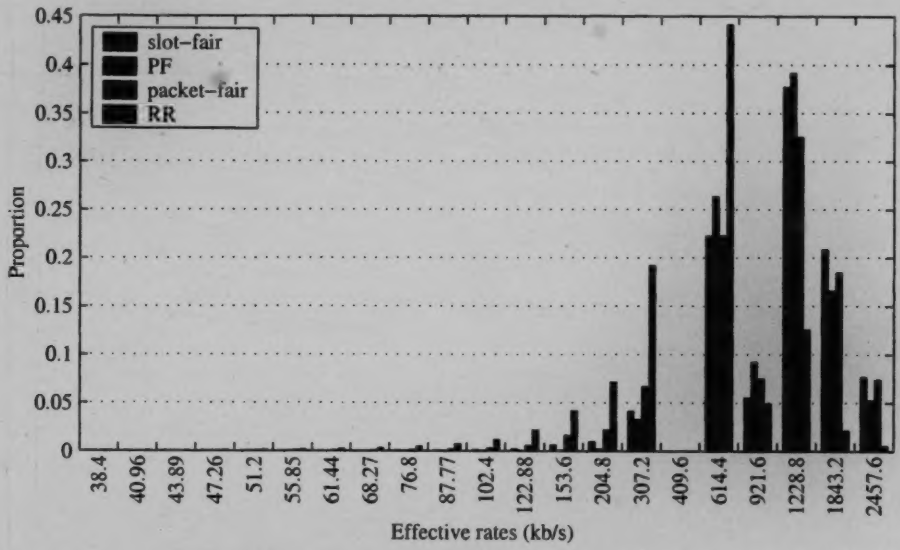


Figure 5.92: Distributions of effective bit rates for packets transmitted under the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector

The next two figures show the distributions of formats and number of transmissions used for correctly received packets.

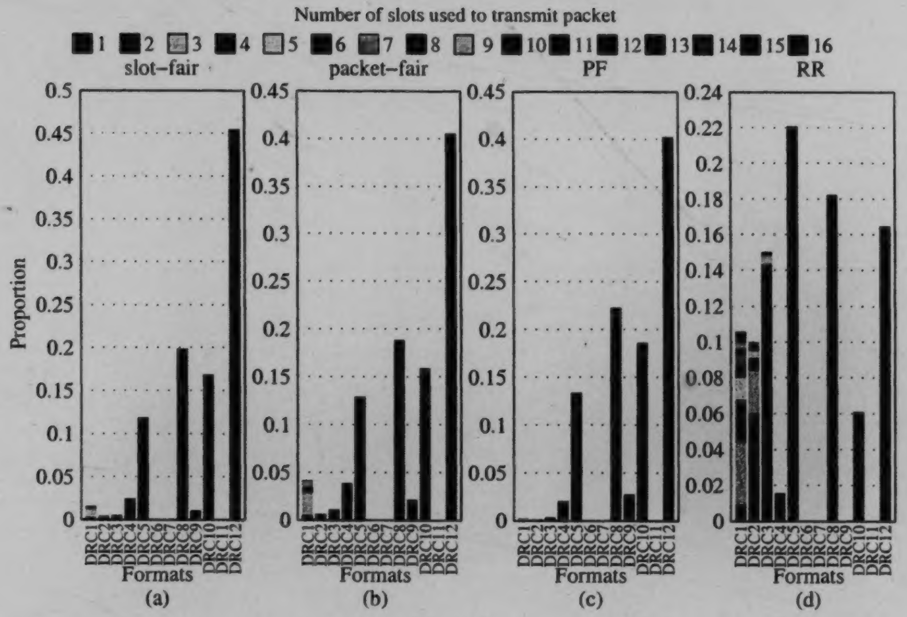


Figure 5.93: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a) slot-wise fair, (b) packet-wise fair, (c) proportionally fair, and (d) round robin scheduling algorithms in the ITU pedestrian A channel with 16 users per sector

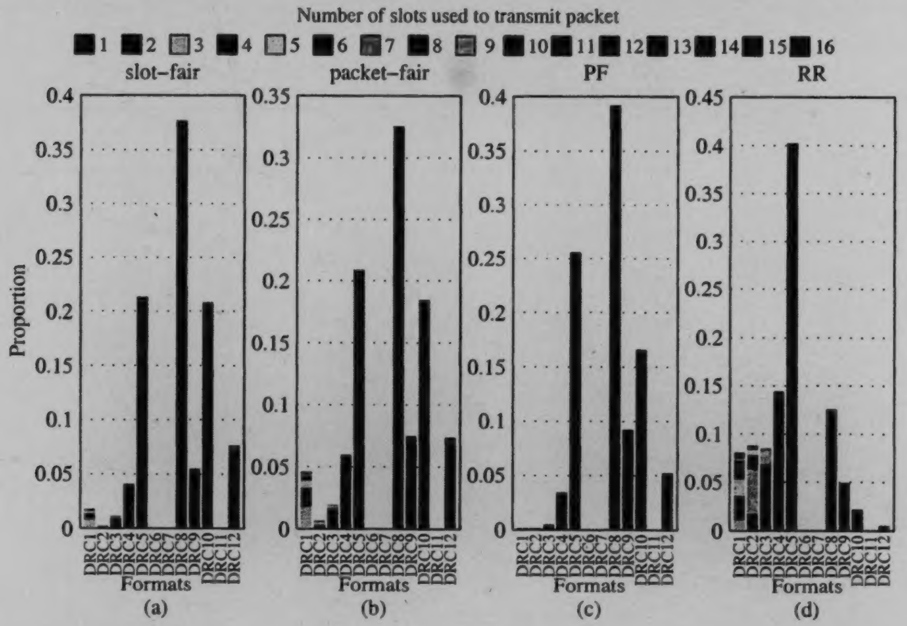


Figure 5.94: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a) slot-wise fair, (b) packet-wise fair, (c) proportionally fair, and (d) round robin scheduling algorithms in the ITU pedestrian B channel with 16 users per sector

Notice that for the slot-fair algorithm, there is a larger proportion of packets transmitted at the highest rates than for the PF algorithm. However, due to the compensation mechanism for users with poorer signals who therefore receive service less frequently, there are also a larger proportion of packets transmitted with the lowest rates. The two extremes seem to cancel each other out, as the average throughputs are about the same. For the packet-fair algorithm, the effective rates of the packets are much higher than for the RR, reflecting the results already seen.

The following two diagrams show the average delay per packet per drop as a function of the number of users per sector for the slot-fair and packet-fair algorithms.

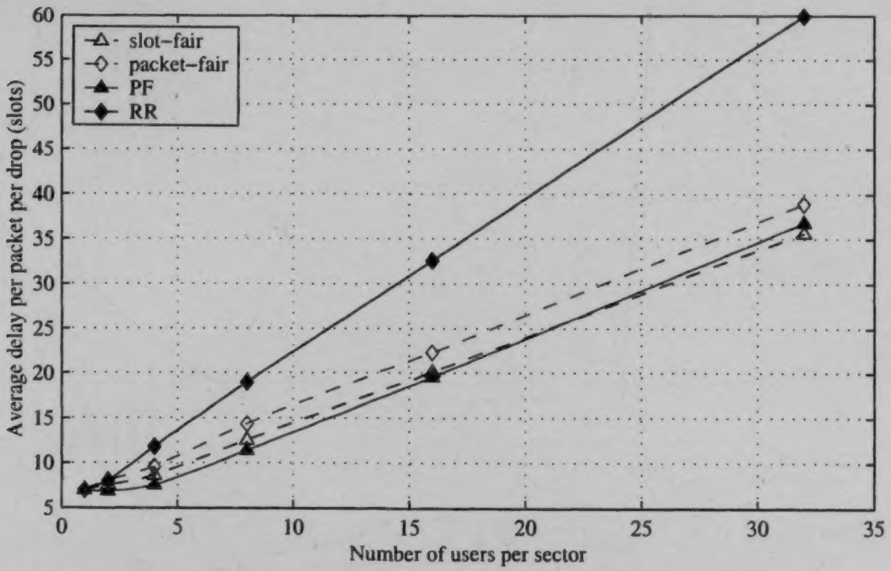


Figure 5.95: Average delay per packet per drop versus number of users per sector for the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian A channel

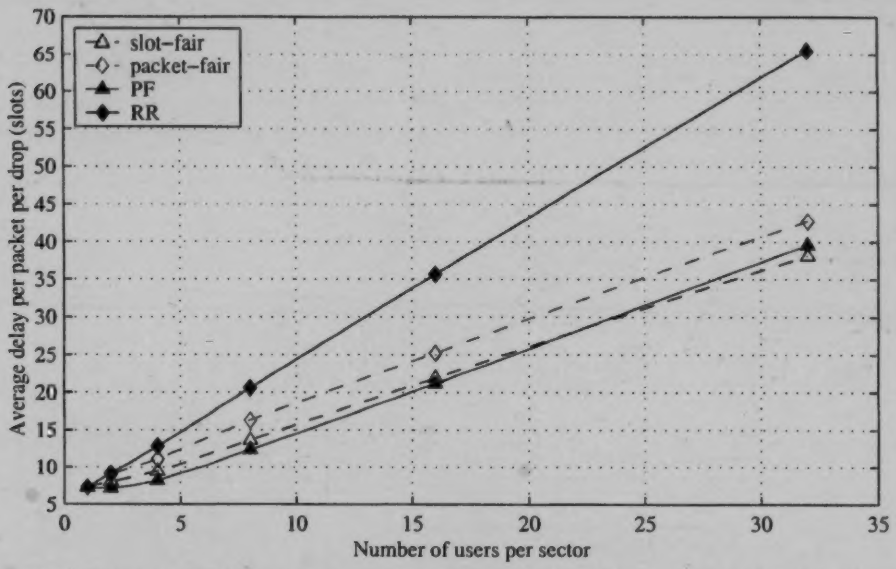


Figure 5.96: Average delay per packet per drop versus number of users per sector for the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian B channel

The delays seen by the slot-fair algorithm and the PF algorithm are fairly similar. There is a crossover that occurs in the curves as there was for the throughput results. The crossover now occurs around 20 to 25 users per sector as opposed to about 16 users per sector as before. For the packet-fair algorithm, the delay is much smaller when compared to the RR algorithm. The reason for the large delay under the RR algorithm is related to not taking channel conditions into account, and is explained in more detail in the section for that algorithm. Clearly, however, the packet-fair algorithm is an improvement over the RR algorithm in terms of delay as well as throughput.

The following two diagrams show the average packet delay per user per drop as a function of the number of users per sector.

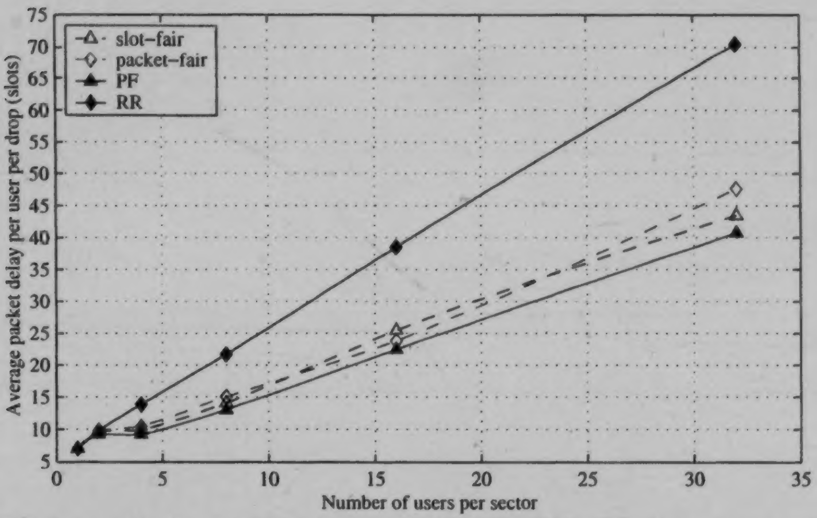


Figure 5.97: Average packet delay per user per drop versus number of users per sector for the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian A channel

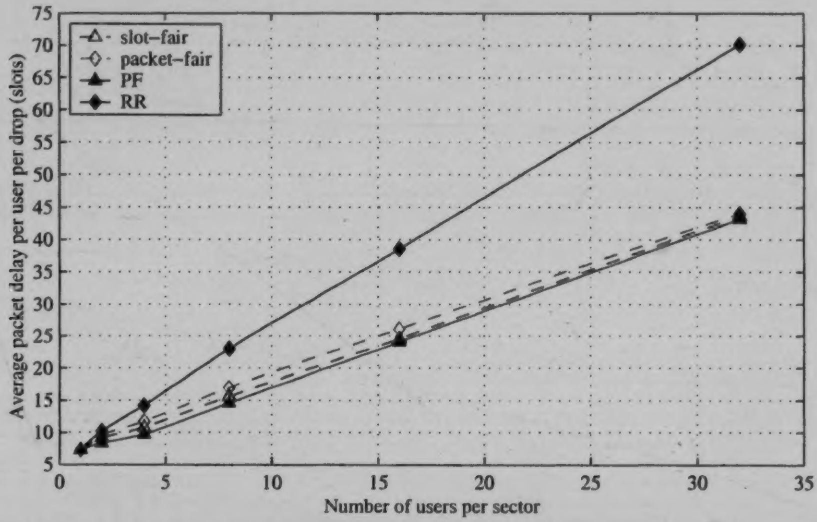


Figure 5.98: Average packet delay per user per drop versus number of users per sector for the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian B channel

In general, the trend for the average packet delay per user is the same as for the average delay per packet. The slot-fair algorithm and the PF algorithm are fairly similar, while the delay of the packet-fair algorithm is larger than those two, but much smaller than that of the RR algorithm. There is a larger difference in the delay between the slot-fair and PF algorithms for larger groups of users in the ITU pedestrian A channel than in the ITU pedestrian B channel. There is also a strange downward bump in the delay of the packet-fair algorithm at 16 users per sector in the ITU pedestrian A channel than cannot be explained.

The following figures show the distributions of delay per packet for the slot-fair and packet-fair algorithms with 16 users per sector.

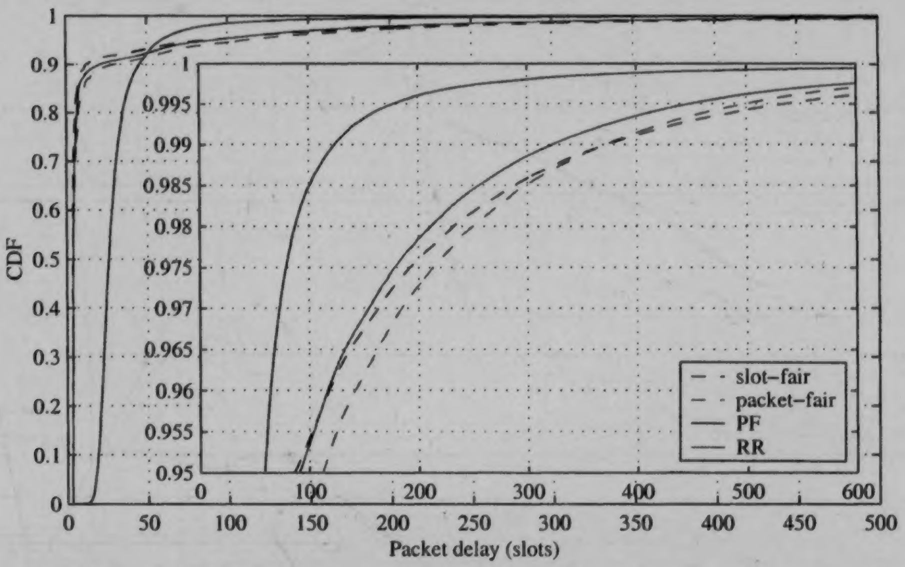


Figure 5.99: Cumulative distribution functions of delays experienced by packets under the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector

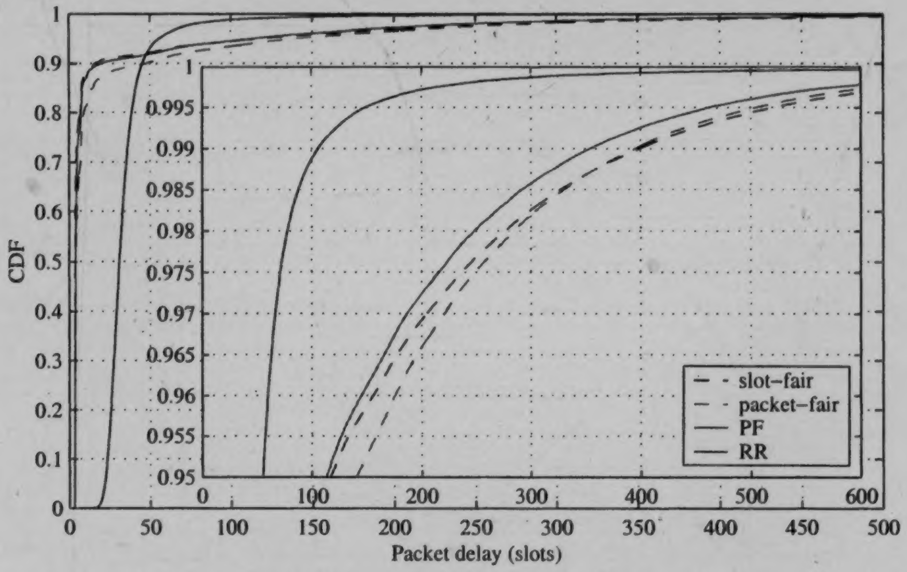


Figure 5.100: Cumulative distribution functions of delays experienced by packets under the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian B channel with 16 users per sector

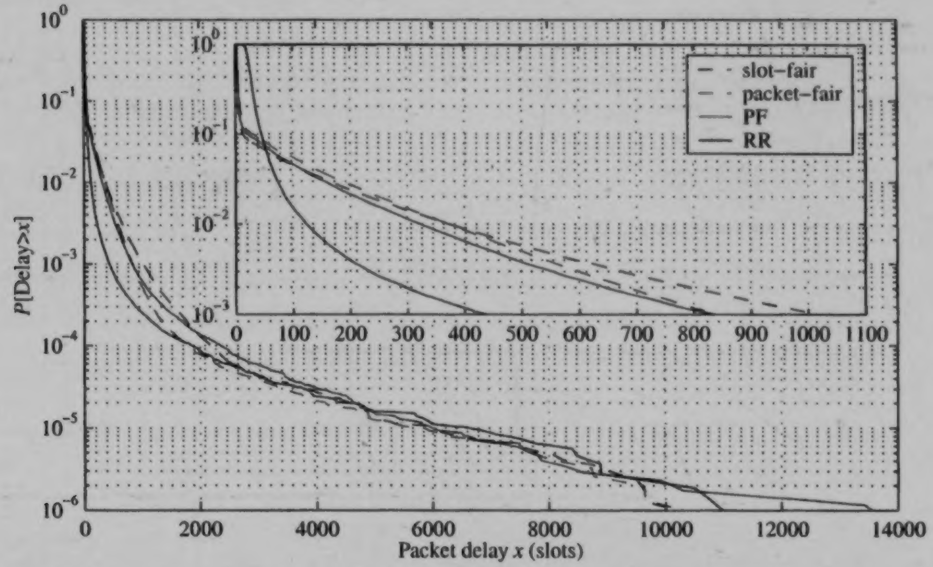


Figure 5.101: Probability of delays experienced by packets being larger than a given value under the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector

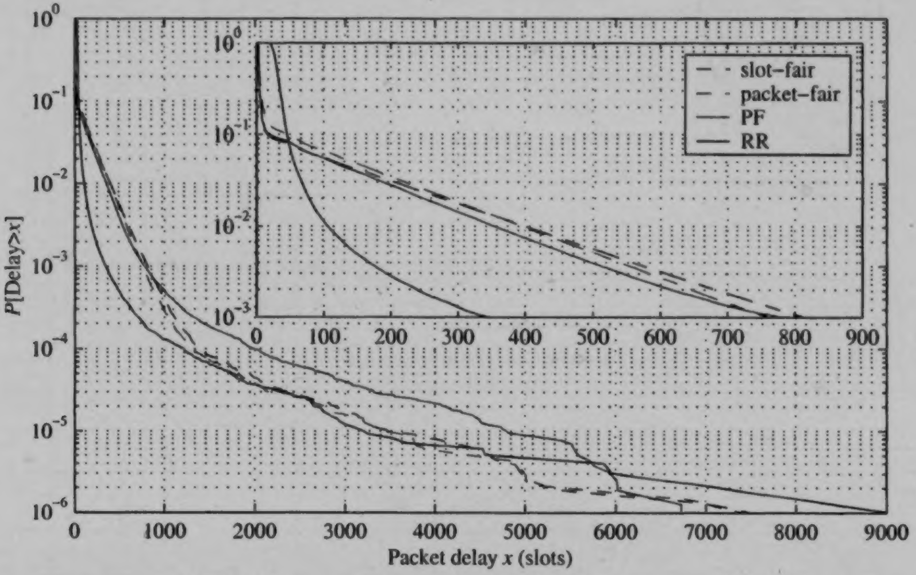


Figure 5.102: Probability of delays experienced by packets being larger than a given value under the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian B channel with 16 users per sector

There again is not a large difference to be observed between the distributions of packet delay for the slot-fair and PF algorithms. The packet-fair distribution is also reasonably close to the other two, although the delays seen are marginally higher. The PF algorithm in the ITU pedestrian B channel seems to have a higher probability of long delays, but for reasonably short delays (e.g.

below 1000 slots), the distributions are quite similar. The largest difference is again the round robin algorithm. It demonstrates a much lower probability of packets with very short delays (i.e. below 50 slots). However, the probability of a delay between 50 and 1000 slots is much lower than for the other three algorithms. For larger delays (i.e. greater than 2000 slots), the algorithms are all much the same. In this region, the delays are more characterized by users who go into outage than by decisions made by the scheduling algorithm.

The following table gives the percentiles and standard deviations of the above distributions.

Table XXV:
Percentiles and Standard Deviations of Packet Delays for the Slot-wise Fair and Packet-wise Fair Scheduling Algorithms in the ITU Pedestrian A and B channels with 16 Users per Sector

Channel	ITU pedestrian A				ITU pedestrian B			
Algorithm Percentile	slot-fair	packet-fair	PF	RR	slot-fair	packet-fair	PF	RR
50th (median)	4	4	4	25.9	4	4	4	30.7
90th	12.4	29.8	20.9	45.2	15.9	46.5	19.6	45.1
95th	86.0	111.7	90.4	58.9	116.8	140.4	113.0	55.4
99th	372.9	371.7	322.6	123.4	399.2	395.4	353.6	105.8
99.5th	535.6	500.0	448.3	177.2	520.3	504.9	461.3	148.4
99.9th	1017.6	842.9	837.9	438.9	813.4	761.5	777.2	344.4
99.95th	1293.2	1019.6	1095.7	655.1	949.1	881.5	998.5	493.2
99.99th	2180.6	1785.9	2345.1	1870.2	1382.2	1333.4	1987.0	1216.4
Std. Dev.	87.73	81.53	80.64	52.06	78.16	77.95	76.93	30.10

The quantitative values reflect the qualitative descriptions of the distributions above. Note the similarity in the values for the standard deviations for the slot-fair, packet-fair, and PF algorithms. This is a further indication of how close all three distributions are to each other.

The following two figures show the distributions of average packet delay per user with 16 users per sector.

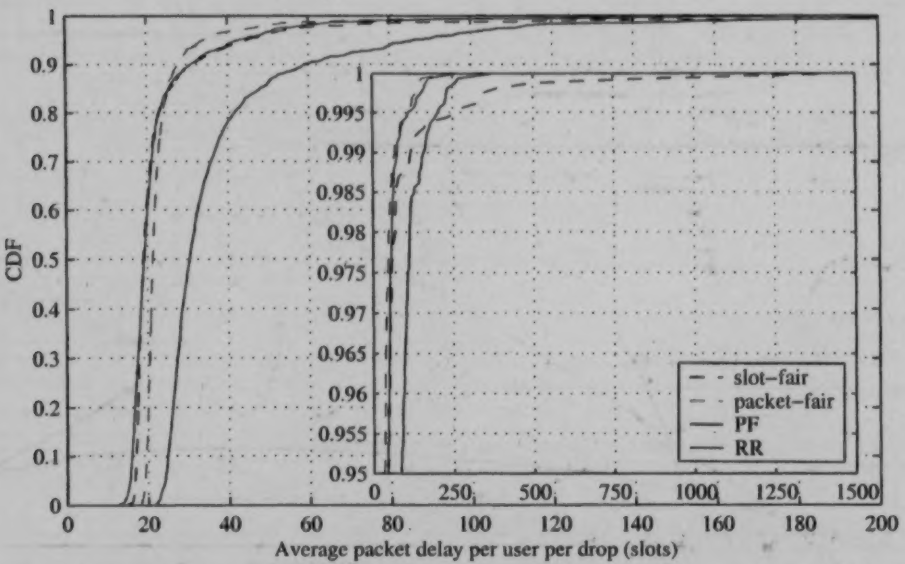


Figure 5.103: Distributions of average packet delay per user for slot-fair and packet-fair scheduling algorithms in ITU pedestrian A channel with 16 users per sector

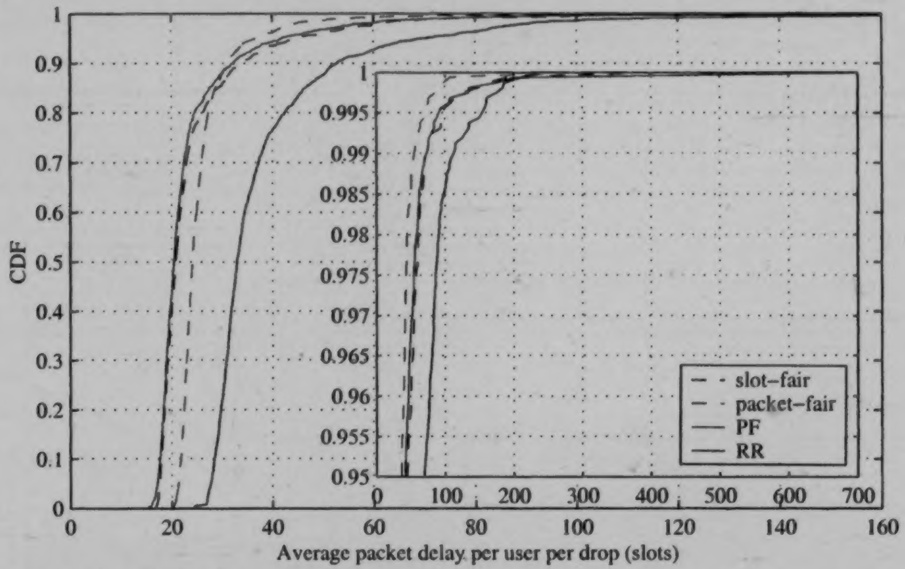


Figure 5.104: Distributions of average packet delay per user for slot-fair and packet-fair scheduling algorithms in ITU pedestrian B channel with 16 users per sector

Once again, the distributions of delays under the slot-fair and PF algorithms are very similar. However, there is an extended tail on the distribution for the slot-fair algorithm in the ITU pedestrian A channel which diverges from the PF algorithm at a delay of about 50 slots. The reason behind this lies in how the algorithms handle the few users in bad channel conditions who intermittently go into outage. In the ITU pedestrian A channel, the signal strength of the users stays relatively constant, so if a user goes into outage, that user tends to stay in outage for an extended period of time. For a user near outage, when that user does receive data, it would be at the lowest possible rate, and therefore would use multiple slots. That can quickly cause a user to receive a large proportion of the slots that have been allocated. As such, the user is then forced to wait until the proportion decreases again. If that user happens to be in outage when this decrease occurs, this would result in an even longer wait, and would make the user fall even further behind in its proportion of packets. When the user can once again support a non-zero rate, it would have high priority to receive packets (likely several in a row). However, once again the packets would be sent at the lowest possible rate. This again could lead to the user receiving too large a proportion of packets, and so the whole cycle would start over again. This problem would only occur for those few users hovering near outage, which can inflate the delay times. Users with stronger signals would share the slots more efficiently, so their delays would not be so large. This effect is not visible in the ITU pedestrian B channel, as the multipath nature means more signal strength variation, and as such users are less likely to be outage for extended periods of time. The effect is also not seen for the packet-fair algorithm, since a single transmission would affect the metric a single time (i.e. 1 packet) instead of multiple times (i.e. up to 16 times for a single packet).

For the packet-fair algorithm, the delays seen are somewhat higher, but there is also a sharp increase in the distribution. Above a threshold of 25 to 30 slots, the packet-fair algorithm actually has the smallest probability of the four algorithms of a delay per user greater than the given value.

The following table gives some percentiles and the standard deviations of the distributions of average packet delay per user.

TABLE XXVI:
PERCENTILES AND STANDARD DEVIATION OF AVERAGE PACKET DELAY PER USER FOR THE SLOT-WISE FAIR AND PACKET-WISE FAIR SCHEDULING ALGORITHMS IN THE ITU PEDESTRIAN A AND B CHANNELS WITH 16 USERS PER SECTOR

Channel	ITU pedestrian A				ITU pedestrian B			
Algorithm / Percentile	slot-fair	packet-fair	PF	RR	slot-fair	packet-fair	PF	RR
10th	17.0	19.6	15.9	24.8	17.9	21.8	17.7	28.4
25th	17.6	20.2	17.0	26.6	18.8	22.7	18.7	30.2
50th (median)	18.9	21.2	19.1	30.3	20.8	24.1	20.4	33.0
75th	21.6	23.2	21.8	37.8	23.8	26.5	23.2	39.0
90th	30.8	27.7	30.9	58.9	34.1	31.0	32.0	52.7
Std. Dev.	50.2	11.7	14.1	26.4	15.2	9.48	21.9	17.7

The larger standard deviation for the slot-fair algorithm in the ITU pedestrian A channel reflects the longer tail of the distribution as discussed earlier. The packet-fair algorithm displays the smallest standard deviation, mirroring the sharp increase in the distribution mentioned above.

The following two figures show the distributions of allocated slots for the slot-fair and packet-fair algorithms with 16 users per sector.

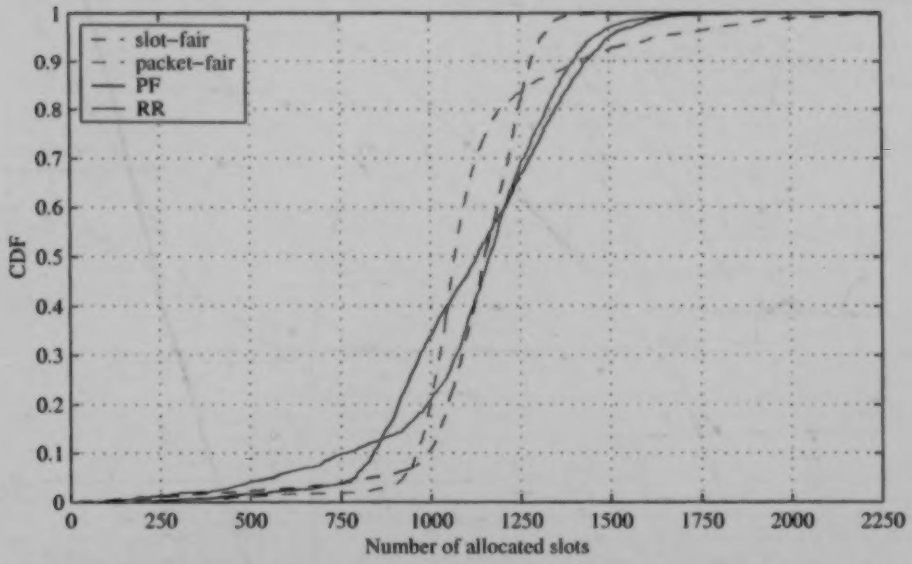


Figure 5.105: Distributions of allocated slots per user per drop for the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian A channel with 16 users per sector

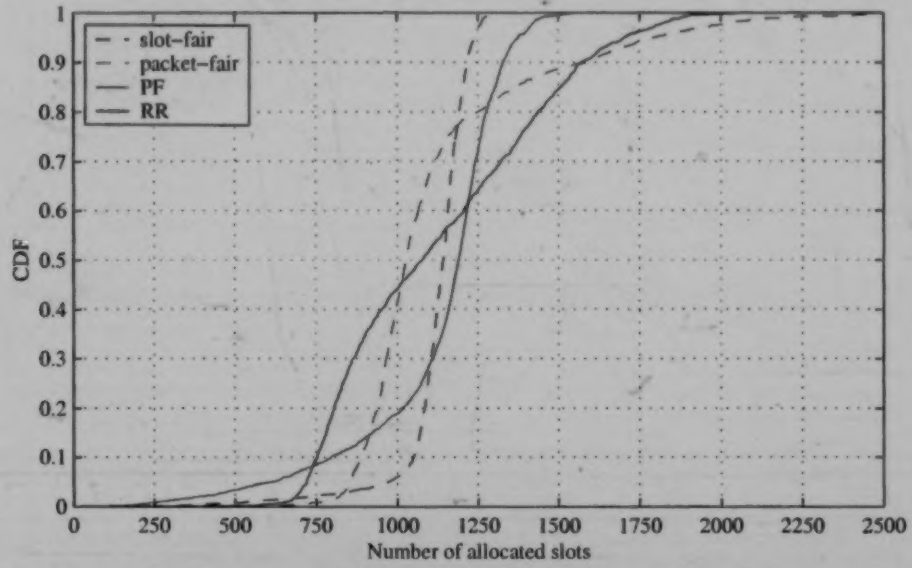


Figure 5.106: Distributions of allocated slots per user per drop for the slot-fair and packet-fair scheduling algorithms in the ITU pedestrian B channel with 16 users per sector

The slot-fair algorithm provides the tightest distribution of slots of the four algorithms. This isn't really a surprise, as the design goal of the algorithm was to equalize the number of slots each user receives. The PF algorithm is the next tightest distribution. The packet-fair algorithm does not provide quite as narrow a distribution as the previous two algorithms, but it is still much tighter than the RR algorithm. The following table shows the percentiles and standard deviations for the slot distributions.

Table XXVII:
Percentiles and Standard Deviation of Slots Allocated per User per Drop for the Slot-wise Fair and Packet-wise Fair Scheduling Algorithms in the ITU Pedestrian A and B Channels with 16 Users per Sector

Channel	ITU pedestrian A				ITU pedestrian B			
Algorithm / Percentile	slot-fair	packet-fair	PF	RR	slot-fair	packet-fair	PF	RR
10th	990	961	766	840	1050	890	797	752
25th	1096	1012	1040	940	1096	955	1075	833
50th (median)	1150	1068	1163	1130	1142	1028	1190	1078
75th	1232	1157	1290	1312	1182	1159	1258	1379
90th	1281	1406	1401	1436	1217	1583	1328	1571
Std. Dev.	183.7	242.6	265.9	247.7	113.8	298.9	231.9	324.4

The numerical values of the standard deviation confirm the tightness of the distributions as described above. The only aberration is that the packet-fair standard deviation is actually smaller than that of the PF algorithm in the ITU pedestrian A channel. The sharp increase in the distribution in the center for the packet-fair algorithm must counteract the longer tail at the upper end of the distribution so that overall, it is in fact somewhat tighter than the PF algorithm.

Overall, the slot-fair algorithm was seen to have a comparable performance to the PF algorithm. The similarity is especially marked at 16 users per sector. The packet-fair algorithm was also shown to be a vast improvement over the RR algorithm in all areas, including throughput, delay, and fairness.

5.3.5. Minimum Performance (minperf)

The goal of the minimum performance algorithm is to maximize the system throughput while simultaneously providing a specified minimum throughput to all users requesting service. The algorithm selects the user with the highest value of $\alpha_k(t)U_k(t)$ for transmission. The term $U_k(t)$ is the same as for the slot-fair and packet-fair algorithms, except that for outage users, $U_k(t) = 0$. Since the metric is a product of the requested rate instead of a sum, there is no problem dealing with users that request a zero rate. The $\alpha_k(t)$ term adapts the users metric based on whether or not the user is receiving the minimum specified throughput, according to the following equation:

$$\alpha_k(t) = \max\left\{\alpha_k(t-1) - a\left[\overline{R}_k(t) - C_k\right], 1\right\} \tag{5.6}$$

The $\overline{R}_k(t)$ term is the average rate as calculated by the PF algorithm, and C_k is the specified minimum throughput, both in units of bits per second. The parameter a affects how quickly the algorithm adapts to changes in the throughputs of the users. A few trial runs with 8 and 32 users per sector in the ITU pedestrian B channel, using perfect multislot prediction to avoid having to iterate, showed that $a = 5\times10^{-5}$ was a good choice that tracked when the throughput of users dropped below the minimum specified value.

As part of the analysis, a variety of rates were used for the minimum specified value of throughput. Specifically, the algorithm was run for $C_k = 5000$ bits per second to 25000 bits per second in steps of 5000 bits per second (i.e. minimum rates of 5, 10, 15, 20, and 25 kb/s). It should also be noted that the maxD algorithm is a special case of this algorithm for which $C_k = 0$.

The following two diagrams show the resulting average throughput per sector as a function of the number of users per sector using the above values.

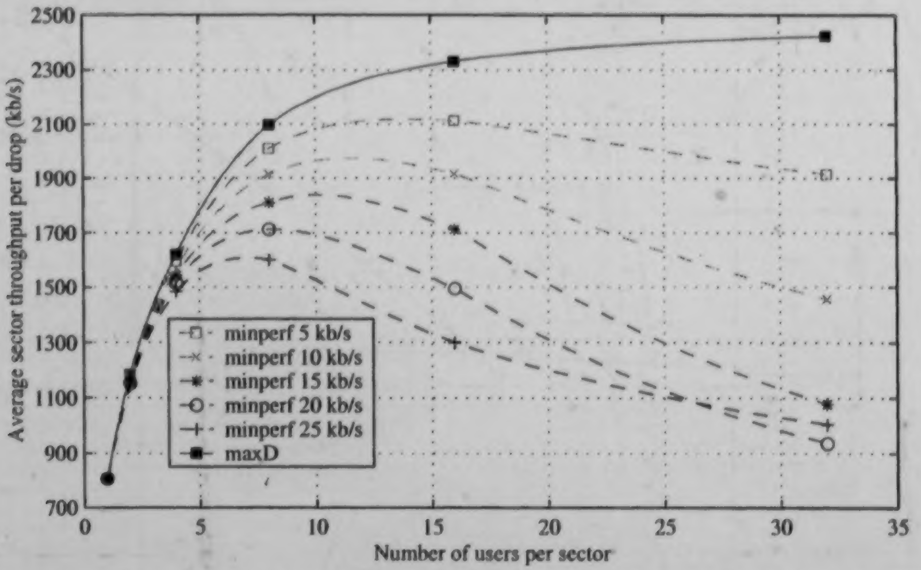


Figure 5.107: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian A channel for the minperf scheduling algorithm

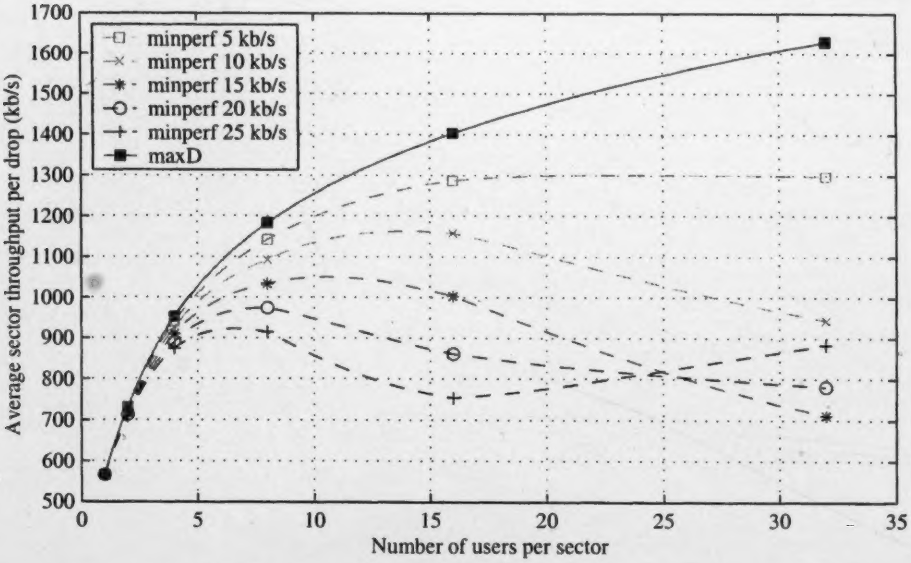


Figure 5.108: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian B channel for the minperf scheduling algorithm

The throughput curves show some interesting results. The throughput increases between 1 user to about 8 to 16 users per sector. This is a direct result of multiuser diversity taking advantage of channel conditions. Beyond the 8 to 16 user-per-sector range, the throughput begins to drop. At this point, the requirement to provide each user with the minimum throughput overrides the desire to transmit to the best user. As the number of users per sector increases, this override occurs more and more often. Because transmissions no longer occur at the optimal time as a result of providing the minimum throughput, the overall throughput per sector consequently drops. At the extreme end of the scale, when there are many users per sector, and the minimum specified throughput is relatively high (e.g. above 16 users per sector and C_k = 20 or 25 kb/s), the rate at which the throughput drops becomes smaller, and in some cases, the throughput can actually increase again. In these extreme cases, the channel capacity is such that the minimum throughput cannot be provided to all users requesting service. In this event, the minimum throughput is only provided to the users with medium to the best channel conditions. Those users with the weakest channel conditions are ignored for the majority of the time, and as such receive throughputs well below the specified minimum. Essentially, this occurs because as the system is serving a few users, the throughput calculation of the other users (both with bad and comparatively good channel conditions) drops. However, when operating near the channel capacity, this means that the throughput of both the users with weak signals and users with stronger signal might drop below the specified minimum. Since the stronger users would request higher rates, they would get a higher priority than the weaker users. Therefore, the weaker users would have to wait longer. In the meantime, the throughput of other stronger users is also dropping, so they would still wind up receiving priority. Overall, the factor $\alpha_k(t)$ would have to increase by a factor of up to 64 (the ratio of the highest to lowest possible non-zero requested rate) in order for that weaker user to receive service. The end result is that since most of the time only the users with the best channel conditions are being served, the throughput may actually increase once again.

The following two graphs show the distributions of throughputs per user per drop for the minperf algorithm with 16 users per sector.

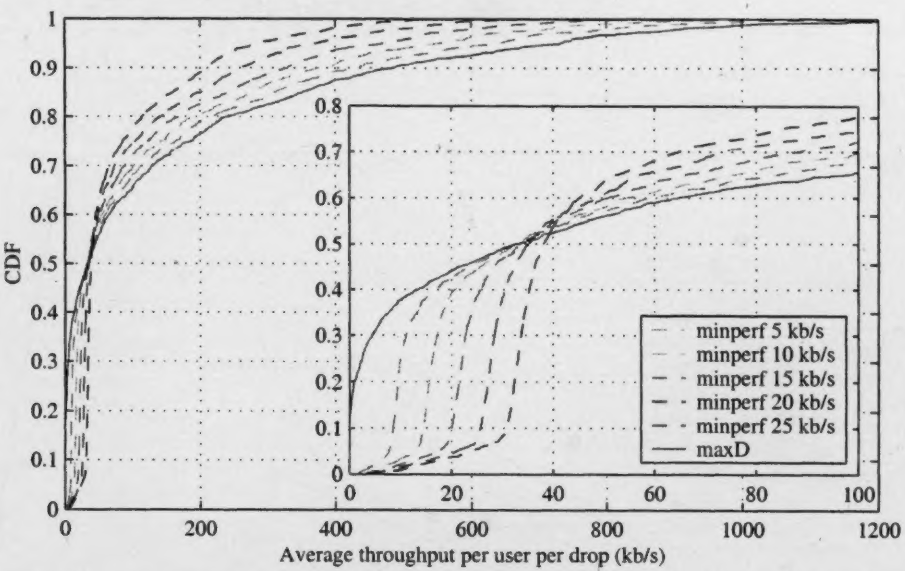


Figure 5.109: Distributions of average throughput per user per drop for the minperf scheduling algorithm in the ITU pedestrian A channel with 16 users per sector

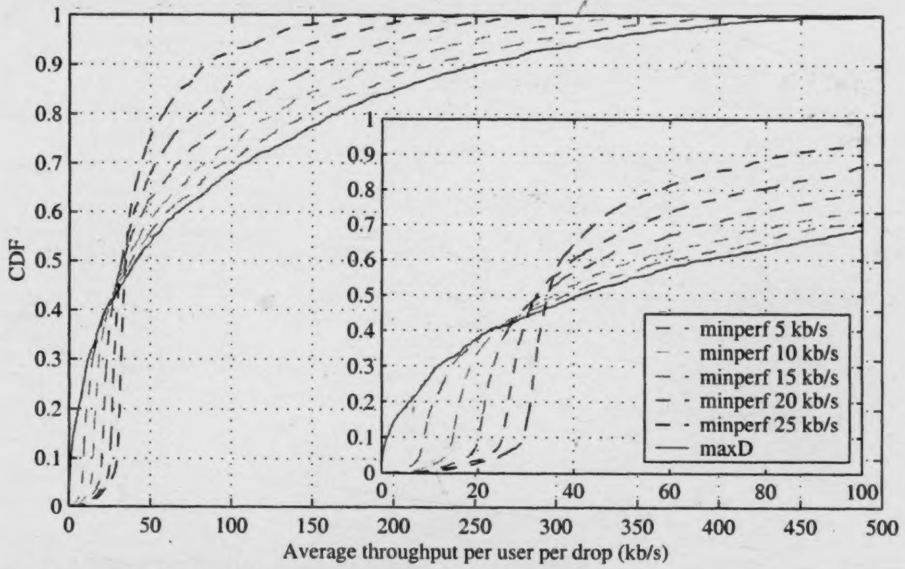


Figure 5.110: Distributions of average throughput per user per drop for the minperf scheduling algorithm in the ITU pedestrian B channel with 16 users per sector

Notice the general shape of the throughput curves. There is a sharp jump in the distributions around the minimum throughput that was specified. After that jump, the shape of the curve is very close to that of the maxD algorithm. This essentially defines how the minperf algorithm works. If a user can normally support a throughput larger than the minimum specified under the maxD algorithm, it will receive that throughput almost as if the system were operating under the maxD algorithm. If the user would normally receive less than the minimum throughput, it will instead receive that minimum and pretty much no more than that minimum. Thus, the throughput is maximized under the minimum throughput constraint. Note that the jumps do not occur exactly at the minimum specified throughputs. This occurs for two reasons. First, when a users throughput is increased to meet the minimum value, it is done so in a sort of quantized manner. The minimum rate at which a user can receive a packet is 38.4 kb/s. Therefore, when increasing the throughput, it will increase by using at least that minimum rate. This will often result in the user then having a throughput greater than the minimum, and so the jump in the distributions occurs at a value large than the minimum. This also explains why the jump is not vertical, but is spread out over several kb/s. Secondly, the calculation of the average rate for the metric was done so using he method for the PF A algorithm. That calculation method does not take into account early terminations. Since the users who receive only the minimum rate have the weakest channel conditions, this means they would most likely receive packets using the lowest rate formats that use the most re-transmissions. These formats, by virtue of the number of slots they use, are the most likely to end early. Therefore, the average rate calculations for the weaker users are likely artificially low due to the calculation method. The algorithm would see that the throughput for a user is below the minimum, even though it may not actually be below (although it would not be very far above that minimum, either.) Therefore, the algorithm would transmit to the users with weaker signals more often than is actually required to meet the minimum specified throughput, and as such, the users receive slightly more than that minimum. Hence, the distributions jump upwards at a throughput larger than the minimum required. Also note that there are still users who cannot support the minimum specified throughput, due to those users going into outage or otherwise being in very poor channel conditions.

The following table shows some statistics of the above distributions. It also gives the probability of a user not receiving the specified minimum throughput.

TABLE XXVIII:
STATISTICS FOR AVERAGE THROUGHPUT PER USER PER DROP DISTRIBUTIONS AND PROBABILITY OF NOT ACHIEVING MINIMUM SPECIFIED THROUGHPUT FOR THE MINIMUM PERFORMANCE SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN A AND B CHANNELS WITH 16 USERS PER SECTOR

Channel	ITU pedestrian A						ITU pedestrian B					
minperf C_k (kb/s)	5	10	15	20	25	0 (maxD)	5	10	15	20	25	0 (maxD)
Mean (kb/s)	132.1	119.8	107.0	93.4	81.2	145.7	80.4	72.3	62.7	53.8	47.1	87.7
Median (kb/s)	32.7	33.0	33.7	35.1	38.1	33.4	38.2	35.9	33.5	32.2	34.5	41.0
Std. Dev. (kb/s)	205.6	178.1	149.5	119.4	91.6	236.8	92.5	77.3	60.3	44.3	30.2	106.5
P[Throughput $<C_k$] (%)	2.06	3.25	3.81	4.56	5.38	-	1.00	1.75	2.63	3.44	4.94	-

The large values of standard deviation indicate how varied the throughputs can be. However, as the minimum throughput specification increases, the standard deviation decreases, as it becomes less likely that a user would receive that throughput based on channel conditions alone (i.e. under the maxD algorithm). Those users therefore receive approximately the minimum specified throughput, resulting in less variation in the throughputs and hence a smaller standard deviation. The mean throughput goes down, as packets are taken away from the best users in order to satisfy the throughput requirements of the weaker users. The median increases, however, as there are more users receiving about the minimum throughput as that minimum increases. The increase in the minimum also increases the median in and of itself. However, the probability of not meeting that minimum also increases, as it simply becomes more likely that the capacity of a user is not sufficient to achieve the minimum.

The following figures show the distributions of effective bit rates for the minperf algorithm with 16 users per sector.

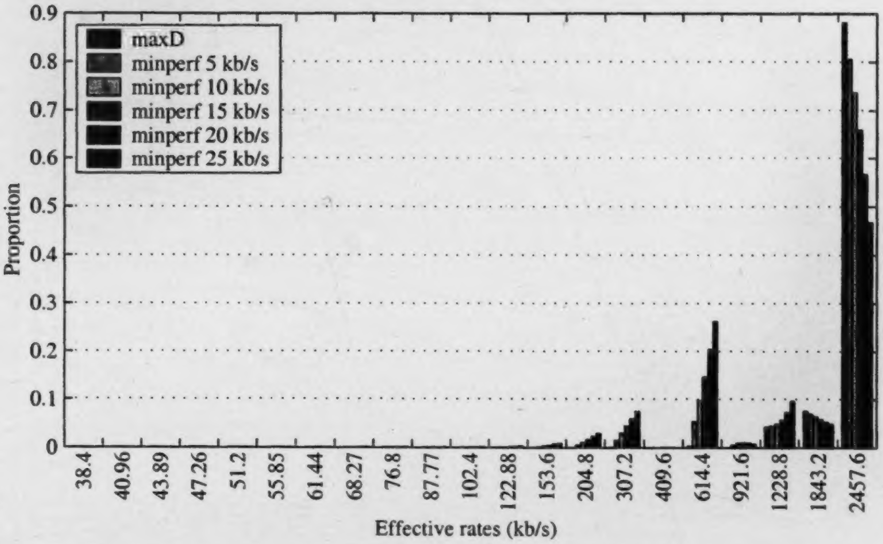


Figure 5.111: Distributions of effective bit rates for packets transmitted under the minperf scheduling algorithm in the ITU pedestrian A channel with 16 users per sector

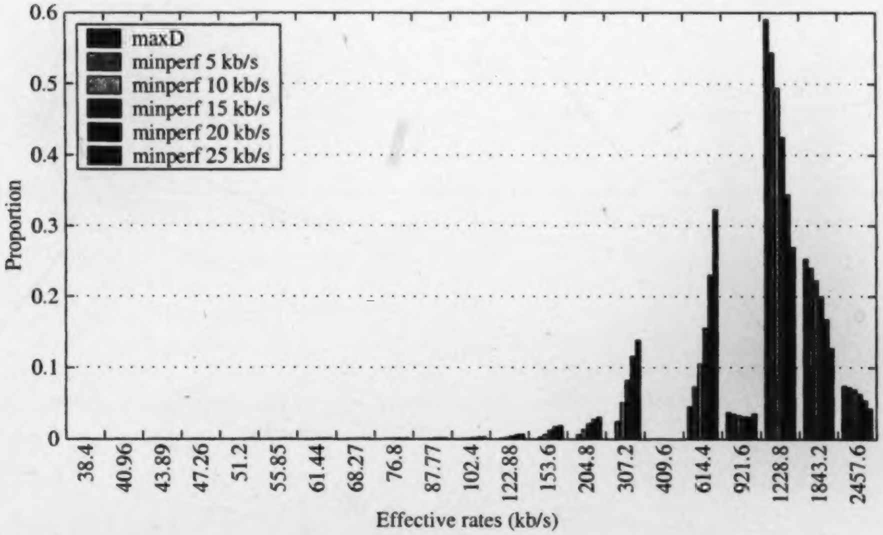


Figure 5.112: Distributions of effective bit rates for packets transmitted under the minperf scheduling algorithm in the ITU pedestrian B channel with 16 users per sector

The next two figures show the distributions of formats and number of transmissions used for correctly received packets under the minperf algorithm with 16 users per sector.

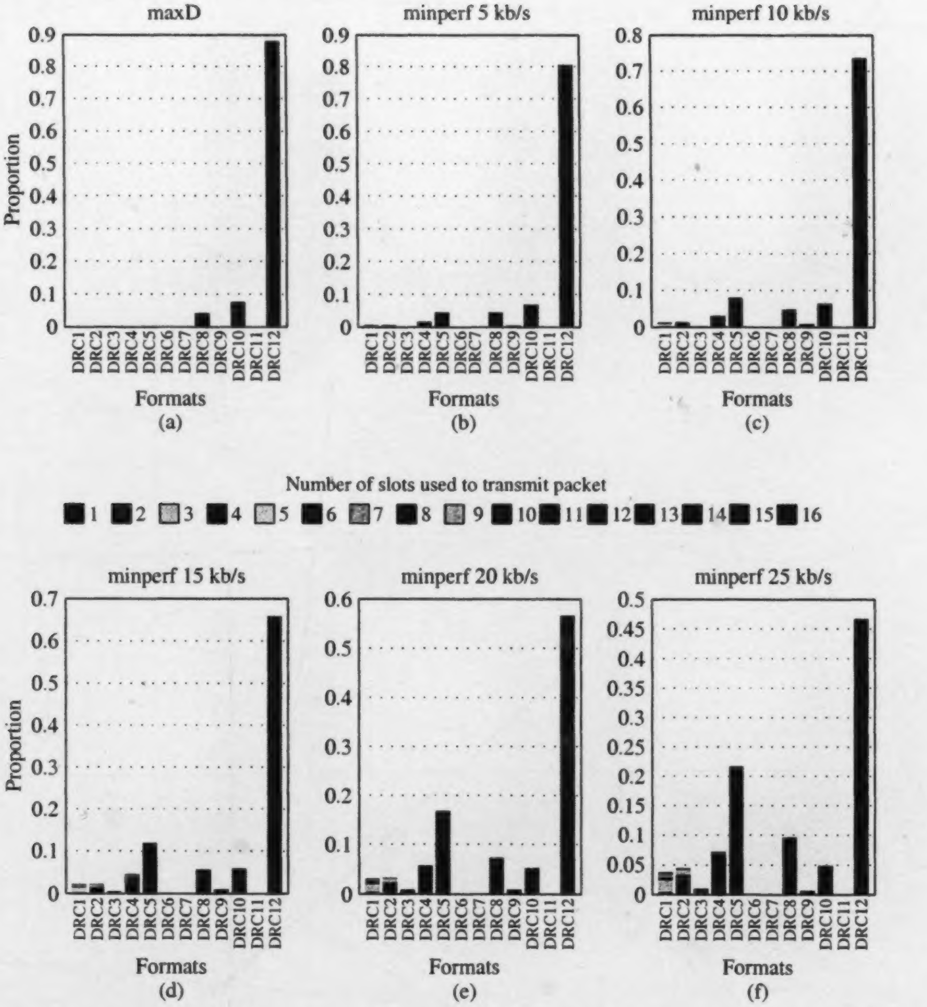
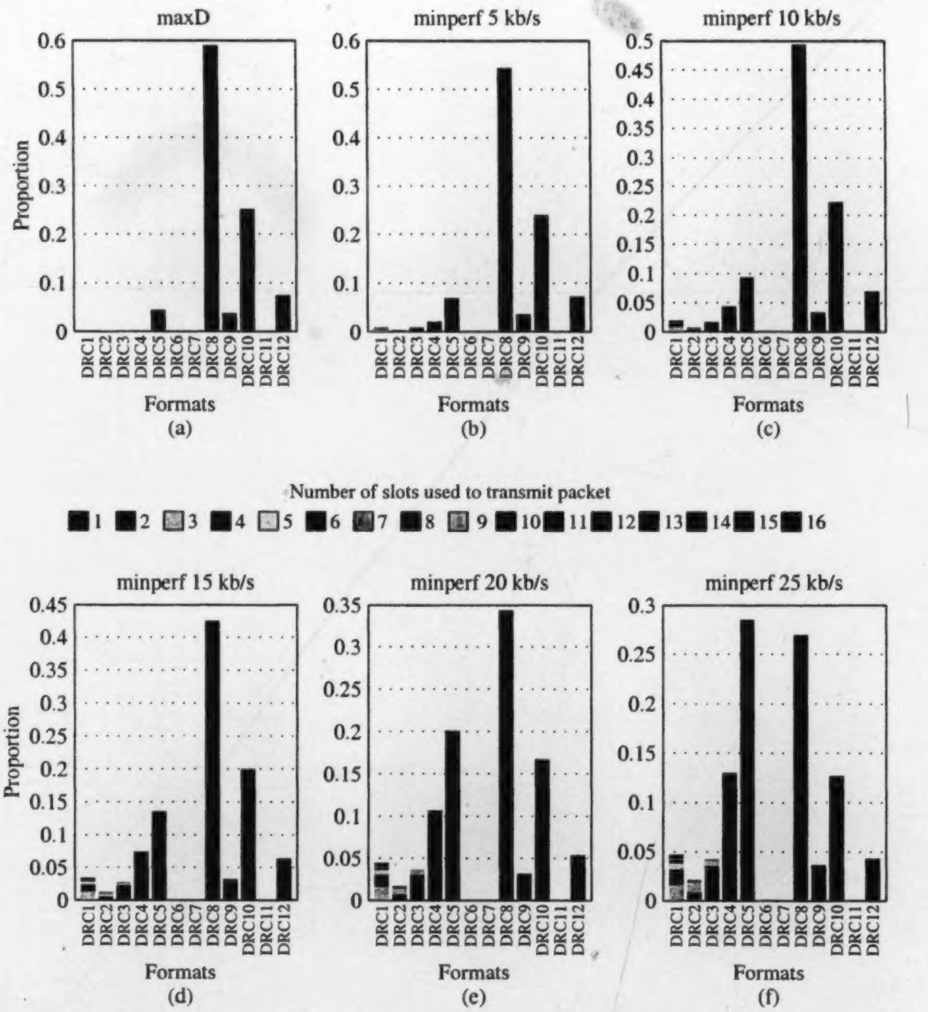


Figure 5.113: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a) maximum rate and minperf algorithm with (b) 5 kb/s, (c) 10 kb/s, (d), 15 kb/s, (e) 20 kb/s, and (f) 25 kb/s minimum throughput specification in the ITU pedestrian A channel with 16 users per sector



(a) (b) (c) (d) (e) (f)

Figure 5.114: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a) maximum rate and minperf algorithm with (b) 5 kb/s, (c) 10 kb/s, (d), 15 kb/s, (e) 20 kb/s, and (f) 25 kb/s minimum throughput specification in the ITU pedestrian B channel with 16 users per sector

As the minimum throughput specification increases, the proportion of packets sent using the highest rates decreases. At the same time, the proportion of packets transmitted using the lowest rate formats increases. The transition in effective rates appears to be reasonably linear with the minimum throughput specification. This is another indicator of how throughput is taken from the users in the best channel conditions in order to satisfy the minimum throughput for the users in poorer channel conditions.

The following diagrams illustrate the average delay per packet per drop as a function of the number of users per sector for the minperf algorithm.

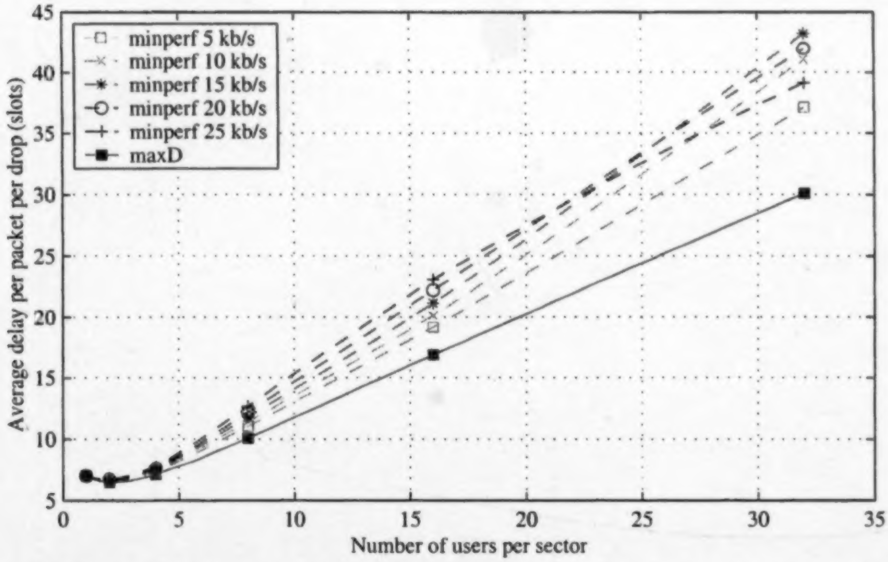


Figure 5.115: Average delay per packet per drop versus number of users per sector for the minperf scheduling algorithm in the ITU pedestrian A channel

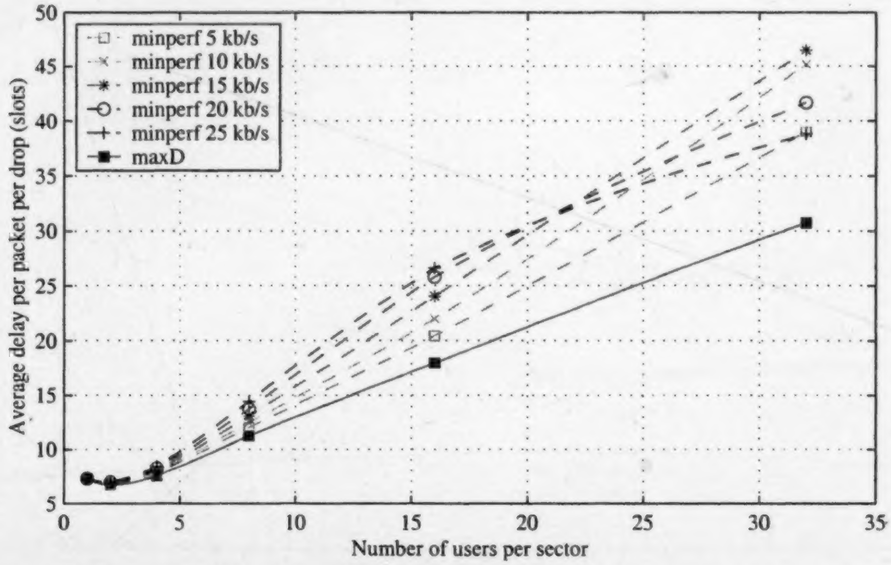


Figure 5.116: Average delay per packet per drop versus number of users per sector for the minperf scheduling algorithm in the ITU pedestrian B channel

The delay per packet curves follow the same kind of pattern that was seen for the average sector throughput curves. There is a general increase in the delay as the minimum specified throughput increases. This corresponds to the decrease in throughput seen earlier. However, as the number of users and the minimum throughput both increased, it was seen earlier that the rate of throughput drop decreased, and in some cases the throughput increased again. The effect is mirrored in the delay curves. For large numbers of users per sector and a high minimum specified throughput, the rate at which the delay increases drops. In the extreme case where the throughput started to curve back towards that of the maxD algorithm, the delay is also seen to be heading back towards the maxD algorithm. This indeed makes sense upon consideration. If the users with the stronger channel conditions begin to gain more priority because their throughputs do not meet the minimum, their packets will be more likely to be transmitted than those belonging to users in poor channel conditions. Those packets would use higher bit rates and fewer slots. Hence, the delay per packet would not increase as rapidly as if the weaker users had higher priority.

The following two graphs show the average packet delay per user per drop as a function of the number of users per sector for the minperf algorithm.

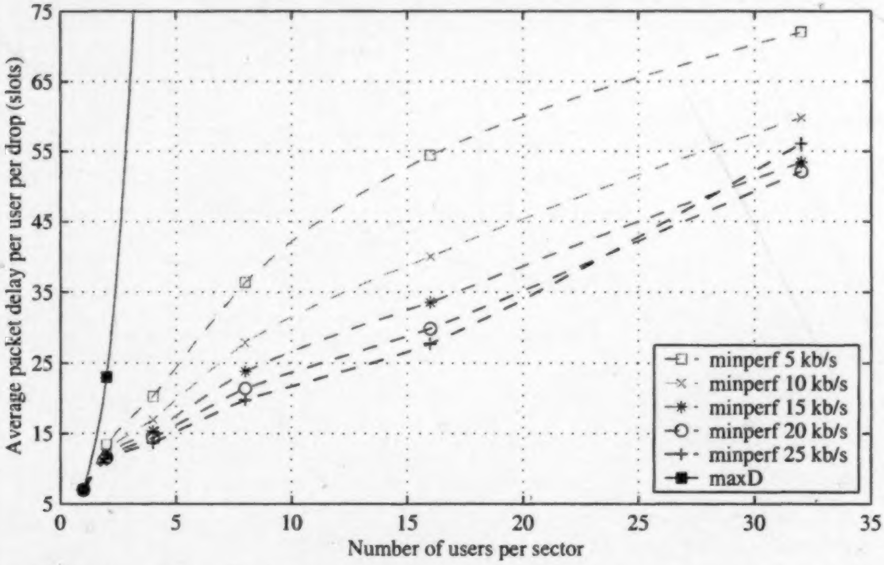


Figure 5.117: Average packet delay per user per drop versus number of users per sector for the minperf scheduling algorithm in the ITU pedestrian A channel

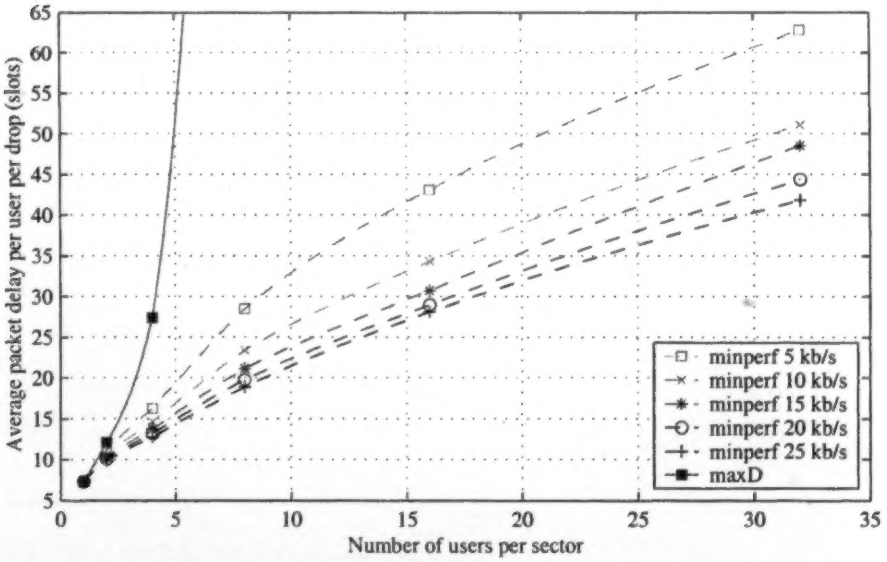


Figure 5.118: Average packet delay per user per drop versus number of users per sector for the minperf scheduling algorithm in the ITU pedestrian B channel

Along with the decrease in throughput and increase in delay per packet, there is also a corresponding decrease in the average packet delay per user with an increasing minimum specified throughput. The amount of decrease becomes less dramatic as the minimum rises, though. The largest change can be seen between not attempting to provide a minimum throughput (i.e. maxD) to attempting to provide some sort of minimum throughput (i.e. minperf 5 kb/s). Thus, by seeing that each user is not starved for service, the extreme nature of the maxD algorithm is removed, and consequently the expected delay seen by a user drops dramatically. Note in the ITU pedestrian A channel, however, that when attempting to provide more throughput to too many users than the channel capacity can support, the delay per user can end up increasing once again, as weaker users start to get ignored. Note also that the increase in the delay isn't really seen for the ITU pedestrian B channel, likely because of the more extensive signal strength variations that allow any given user to be chosen more often and prevent lengthy outages.

The following diagrams show the distributions of delay per packet for the minperf algorithm with 16 users per sector.

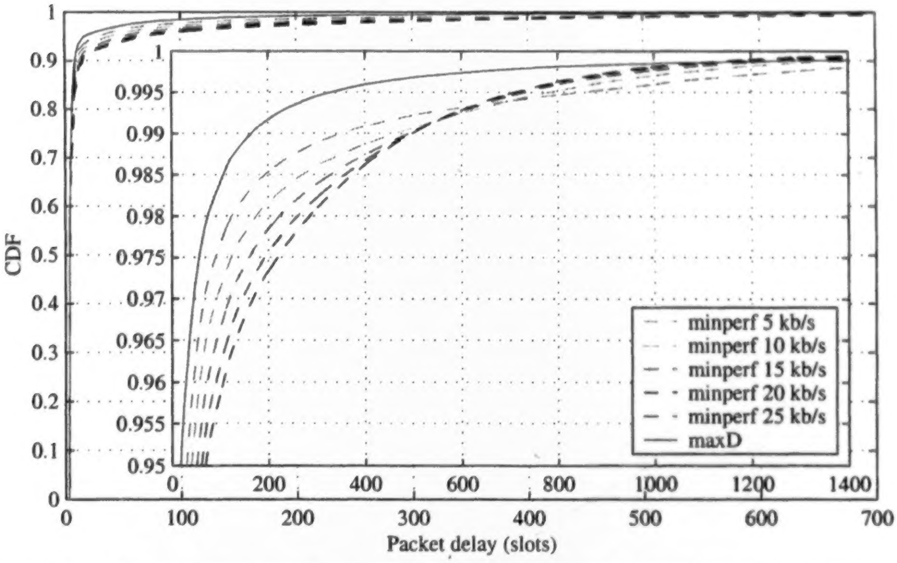


Figure 5.119: Cumulative distribution functions of delays experienced by packets under the minperf scheduling algorithm in the ITU pedestrian A channel with 16 users per sector

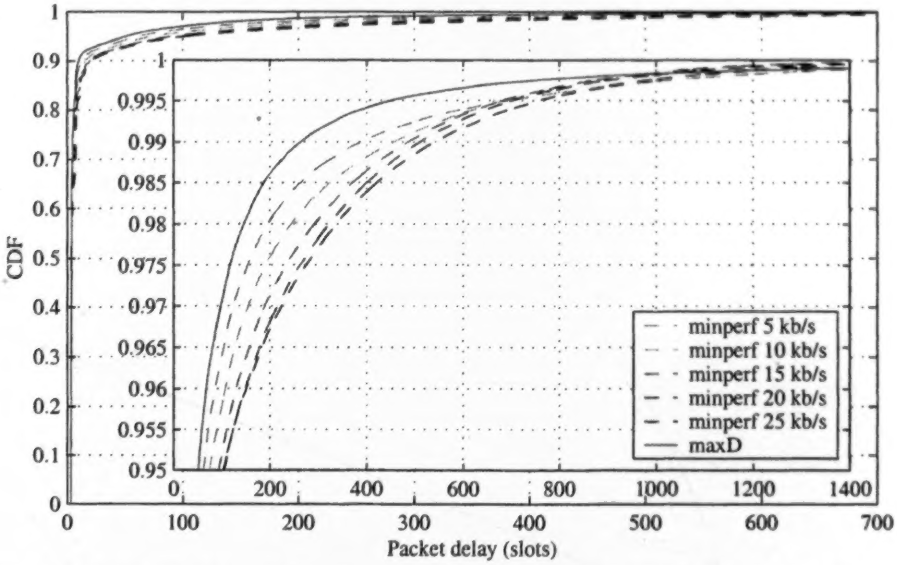


Figure 5.120: Cumulative distribution functions of delays experienced by packets under the minperf scheduling algorithm in the ITU pedestrian B channel with 16 users per sector

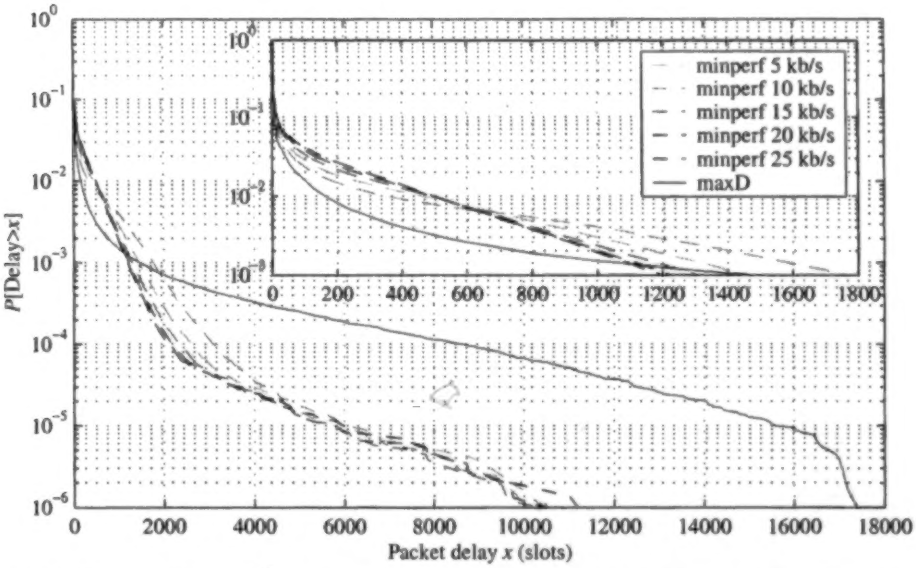


Figure 5.121: Probability of delays experienced by packets being larger than a given value under the minperf scheduling algorithm in the ITU pedestrian A channel with 16 users per sector

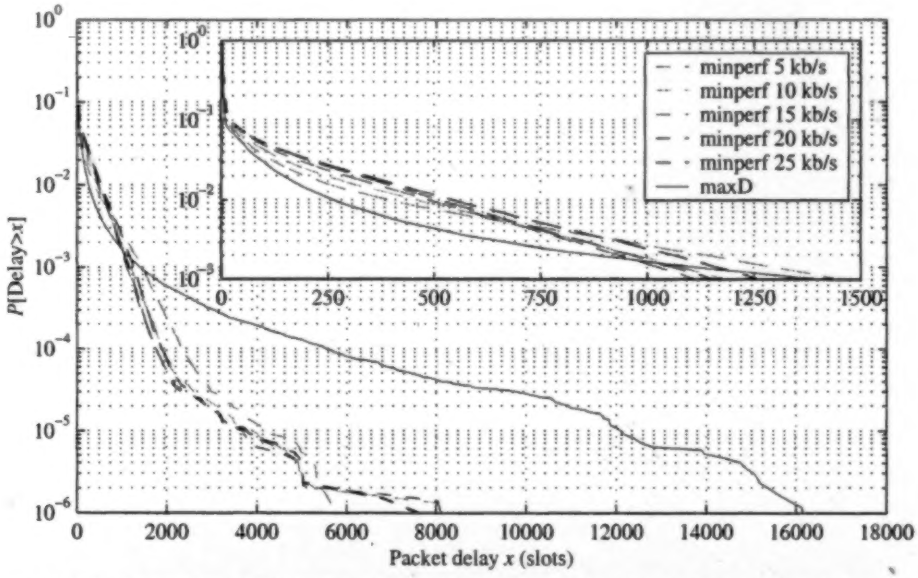


Figure 5.122: Probability of delays experienced by packets being larger than a given value under the minperf scheduling algorithm in the ITU pedestrian B channel with 16 users per sector

As the specified minimum throughput increases, there is a decreasing probability of packets experiencing very short delays. Some of the packets that normally might be sent back-to-back under the maxD algorithm are instead delayed for a time in order to increase the throughput of weaker users. However, there is similarly a decrease in the probability of packets with large delays, since those weaker users are served more often.

The following table shows the percentiles and standard deviations of the packet delays.

TABLE XXIX:
PERCENTILES AND STANDARD DEVIATIONS OF PACKET DELAYS FOR THE MINIMUM PERFORMANCE SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN A AND B CHANNELS WITH 16 USERS PER SECTOR

Channel	ITU pedestrian A						ITU pedestrian B					
C_k (kb/s) / Percentile	5	10	15	20	25	0 (maxD)	5	10	15	20	25	0 (maxD)
50th (median)	4	4	4	4	4	4	4	4	4	4	4	4
90th	10.1	11.5	12.5	15.0	15.6	8.9	12.0	15.6	19.6	21.7	19.9	9.9
95th	29.8	39.9	50.7	60.3	68.4	17.3	62.2	75.0	93.2	105.4	103.5	51.1
99th	351.1	462.3	490.2	492.9	498.1	164.7	389.6	456.7	485.6	511.1	542.1	263.2
99.5th	849.8	782.9	733.0	716.3	711.0	329.3	691.7	685.1	680.8	701.6	757.8	449.6
99.9th	1746.7	1453.1	1307.5	1226.4	1201.1	1476.4	1446.1	1183.6	1112.3	1152.9	1257.3	1399.0
99.95th	2129.0	1759.2	1572.1	1483.6	1427.9	2835.4	1709.0	1404.3	1288.0	1327.2	1435.5	2245.7
99.99th	3097.6	2610.6	2332.4	2210.9	2115.9	8673.6	2335.5	1917.2	1759.0	1749.2	1885.3	5549.0
Std. Dev.	120.51	109.00	103.51	101.98	102.31	168.80	100.27	94.03	94.37	98.57	105.47	132.28

The percentile delays mirror the trend discussed above. As the minimum specified throughput increases, the delay of the lower percentiles (i.e. up to about the 99th percentile) increases, showing decreased likelihood of very short delays. For the upper percentiles (i.e. above the 99.9th percentile), the delay tends to decrease instead. The notable exceptions are the 20 and 25 kb/s cases in the ITU pedestrian B channel. It was already seen how the average delay per packet curves tended to bend back towards the maxD curve. Therefore, it makes sense for the distributions of delay per packet to also tend back towards that of the maxD algorithm. In other words, the percentiles could be expected to (and do) increase. Note that the standard deviation also increases in those two cases, further indicating that the distribution is becoming more dispersed like the maxD algorithm.

The following two graphs show the distributions of average packet delay per user for the minperf algorithm with 16 users per sector.

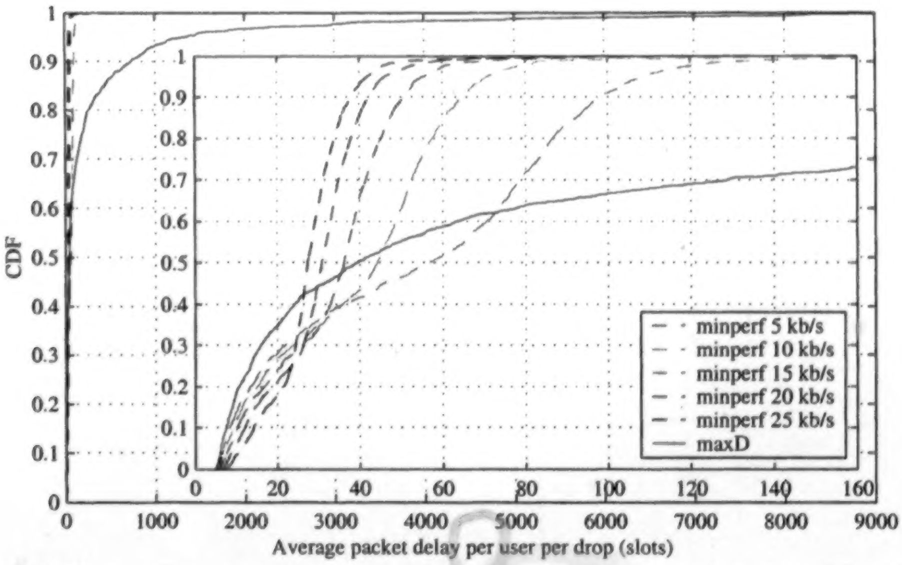


Figure 5.123: Distributions of average packet delay per user for the minperf scheduling algorithm in ITU pedestrian A channel with 16 users per sector

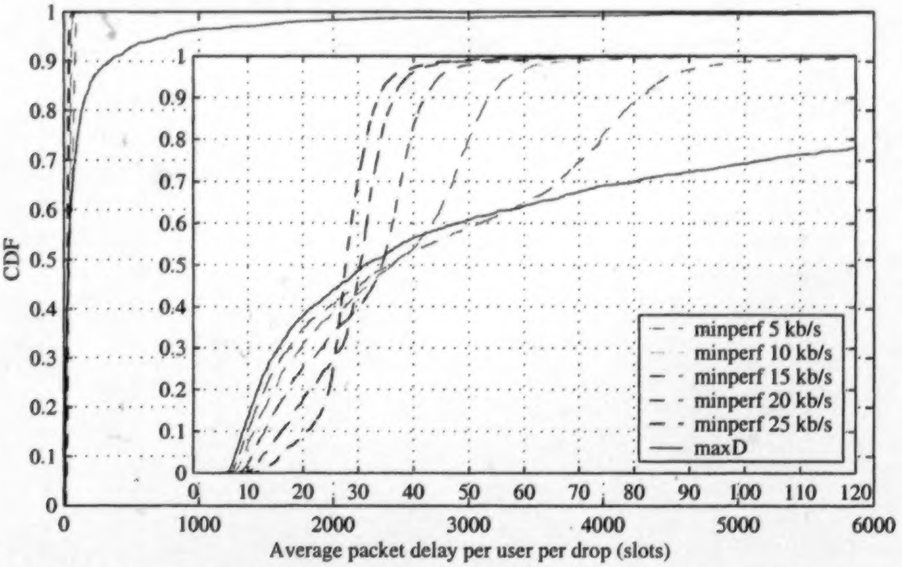


Figure 5.124: Distributions of average packet delay per user for the minperf scheduling algorithm in ITU pedestrian B channel with 16 users per sector

The distributions tend to become narrower as the minimum throughput increases. As the minimum increases, it is more likely that a user will receive approximately that minimum throughput. If many users are receiving similar throughputs, it can be expected that the delays they experience should also be similar. The distributions seem to confirm that notion. Another interesting observation is that, particularly in the lower minimum throughput cases, the distributions tend to rise in two steps. That is, there is an initial increase in the distribution at low values of delay, but there is a second sharp increase in the distribution at intermediate delay levels. The initial rise accounts for the users in the best channel conditions that can support high data rates and therefore low delays. The second rise is attributed to the group of users receiving the minimum throughput, all of whom experience a similar delay.

The following table gives some percentiles and the standard deviations of the above distributions.

TABLE XXX:
PERCENTILES AND STANDARD DEVIATION OF AVERAGE PACKET DELAY PER USER FOR THE MINIMUM PERFORMANCE SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN A AND B CHANNELS WITH 16 USERS PER SECTOR

Channel	ITU pedestrian A						ITU pedestrian B					
C_k (kb/s) / Percentile	5	10	15	20	25	0 (maxD)	5	10	15	20	25	0 (maxD)
10th	8.5	9.3	10.3	12.1	14.2	7.9	10.1	11.2	13.0	16.1	20.8	9.1
25th	16.8	18.6	21.0	23.3	23.4	13.4	14.8	16.6	19.8	24.1	25.4	13.4
50th (median)	56.7	44.2	36.2	30.8	27.4	39.3	35.7	36.2	34.3	30.4	27.8	31.3
75th	83.1	54.8	43.1	36.4	32.4	186.4	70.4	48.4	38.6	33.4	30.7	102.7
90th	97.7	64.8	50.8	42.6	38.0	725.9	81.2	53.5	42.1	36.5	33.8	337.9
Std. Dev.	38.5	22.8	16.3	12.2	10.1	1102.1	31.6	19.5	14.7	10.8	9.11	710.5

The decrease in standard deviation with increasing minimum specified throughput confirms the narrowing of the distribution mentioned above.

The following two graphs show the distributions of allocated slots per user per drop for the minperf algorithm with 16 users per sector.

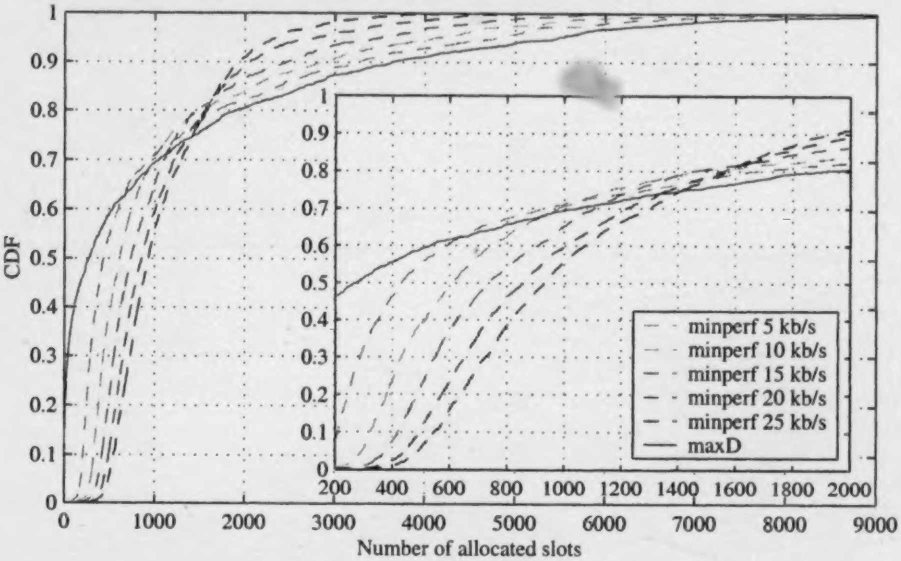


Figure 5.125: Distributions of allocated slots per user per drop for the minperf scheduling algorithm in the ITU pedestrian A channel with 16 users per sector

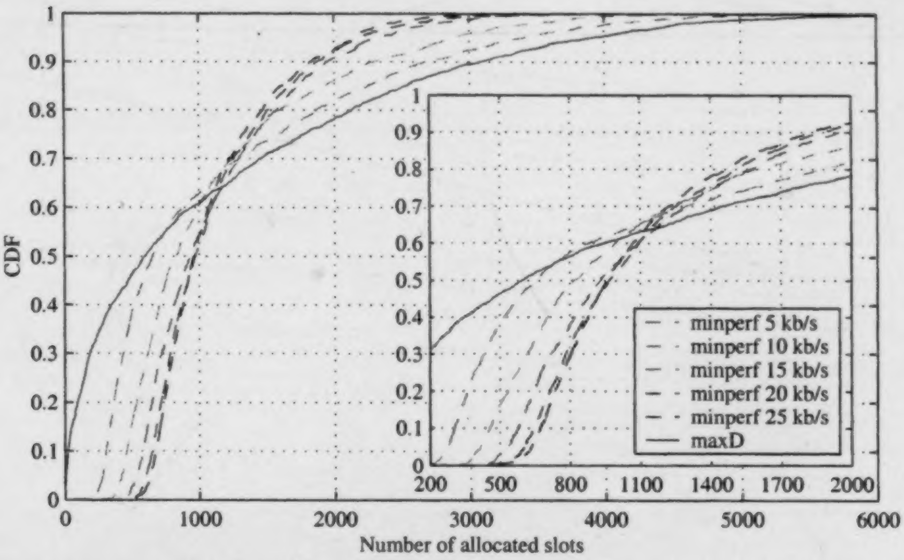


Figure 5.126: Distributions of allocated slots per user per drop for the minperf scheduling algorithm in the ITU pedestrian B channel with 16 users per sector

As with the delay per user, the slot allocation distributions also become narrower with an increasing minimum throughput specification. The reason why is also similar. As the minimum increases, more users receive only about that minimum throughput. With similar throughputs, their other statistics, such as delay per user and number of allocated slots also become similar. Hence, the distributions become tighter as the minimum increases.

The following table shows some percentiles and the standard deviations of the above slot allocation distributions.

TABLE XXXI:
PERCENTILES AND STANDARD DEVIATION OF SLOTS ALLOCATED PER USER PER DROP FOR THE MINIMUM PERFORMANCE SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN A AND B CHANNELS WITH 16 USERS PER SECTOR

Channel	ITU pedestrian A						ITU pedestrian B					
C_k (kb/s) Percentile	5	10	15	20	25	0 (maxD)	5	10	15	20	25	0 (maxD)
10th	210	324	419	493	546	0	305	458	578	647	673	21
25th	264	406	519	609	681	22	390	574	676	751	769	140
50th (median)	434	609	737	858	933	275	654	817	930	968	960	597
75th	1297	1168	1243	1375	1417	1433	1576	1424	1416	1379	1320	1764
90th	3278	2841	2470	2040	1939	3628	2704	2296	1948	1872	1836	3085
Std. Dev.	1499.7	1250.5	1007.8	782.0	621.2	1794.6	1028.3	786.6	582.6	497.6	501.5	1280.3

A decreasing standard deviation with increasing minimum throughput is once again seen. There is only a minor exception in the ITU pedestrian B channel, where the standard deviation at 25 kb/s is almost the same (albeit a small amount higher) than at 20 kb/s.

Overall, the minperf algorithm was generally shown to increase fairness as the minimum specified throughput increased. The average sector throughput would increase with an increase with the number of users per sector for a time as multiuser disversity was exploited, then decrease as the minimum throughput requirement started to override the channel conditions. The throughput and the average delay per user decreased while the average delay per packet increased with an increasing minimum. The distributions also tended to become tighter as the minimum increased. In the extreme cases, with large numbers of users per sector and high values for the minimum, the results could tend back towards that achieved using the maxD algorithm. The rate of decrease in throughput and increase in delay with increasing numbers of users would slow and in a few cases reverse direction. In any event, the minperf algorithm, in attempting to provide a given minimum throughput to each user, was much more fair in comparison to the maxD algorithm, which does not attempt to provide a minimum throughput (or alternatively attempts to provide a minimum throughput of 0 kb/s).

5.3.6. Minimum Performance ("PF B" method)

When discussing the proportionally fair scheduling algorithm, there were two methods analyzed to calculate the average rate. The first method ("PF A") did not take early terminations into account, whereas the second method ("PF B") did. It was found that the difference between the two methods was negligible when used in the context of the PF algorithm. Two reasons for this were given. First, a large portion of the packets were sent using formats where only one slot is used for transmission, so early terminations were not possible. Second, it was speculated that for those packets where early terminations did occur, the average rate was raised by the same proportion for all users. Therefore, in a metric that multiplies or divides by the average rate (such as in the PF algorithm), the user that was relatively best would not change.

The second reason is no longer true in the context of the minperf algorithm. The metric for the minperf algorithm involves a subtraction from the average rate instead of a division by it. Hence, it is possible that the user who is relatively best may now change. Therefore, the simulations for the minperf algorithm were redone using the "PF B" method to calculate the average rate in the metric.

The following two figures show the resulting average sector throughput as a function of the number of users per sector.

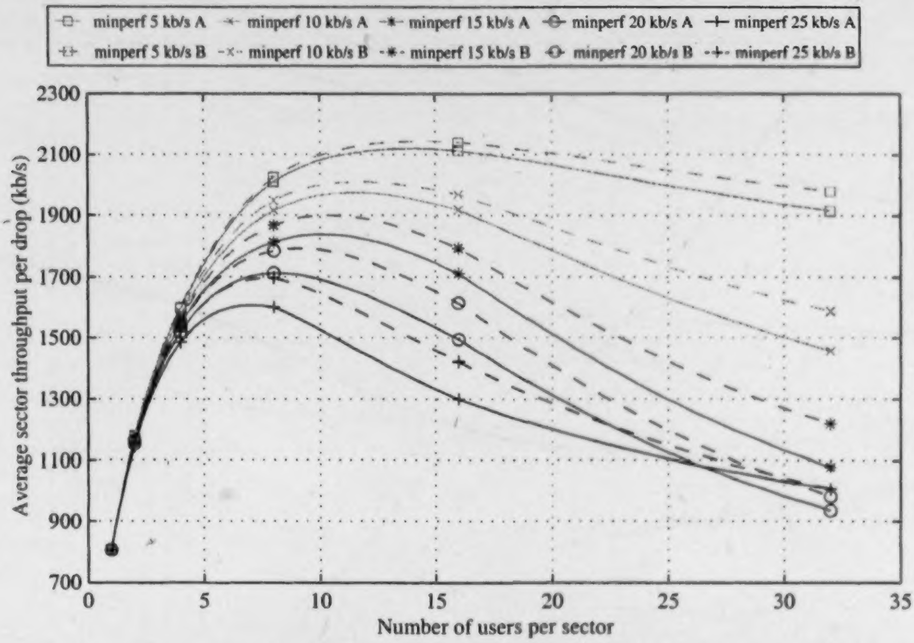


Figure 5.127: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian A channel for the minperf scheduling algorithm using the "PF B" method to calculate the average rate

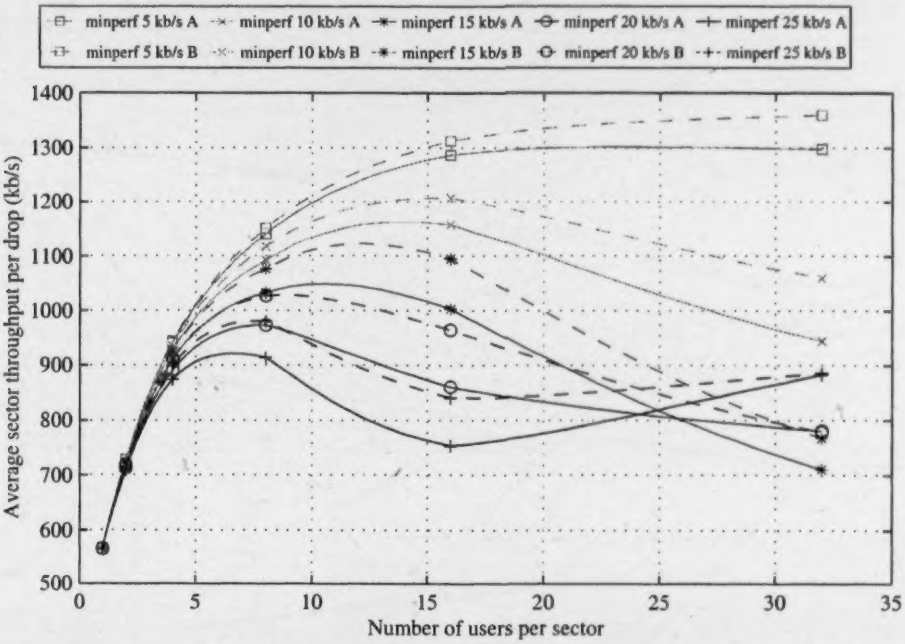


Figure 5.128: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian B channel for the minperf scheduling algorithm using the "PF B" method to calculate the average rate

As can be seen from the above diagrams, unlike with the PF algorithm, there is a significant increase in the throughput achieved when moving from the "PF A" method to the "PF B" method to calculate the average rate. The increase appears to be more pronounced the higher the value of the specified minimum throughput is. The explanation for this is more readily available when the distributions of the average throughput per user per drop are examined. These are shown in the next two figures for the case of 16 users per sector.

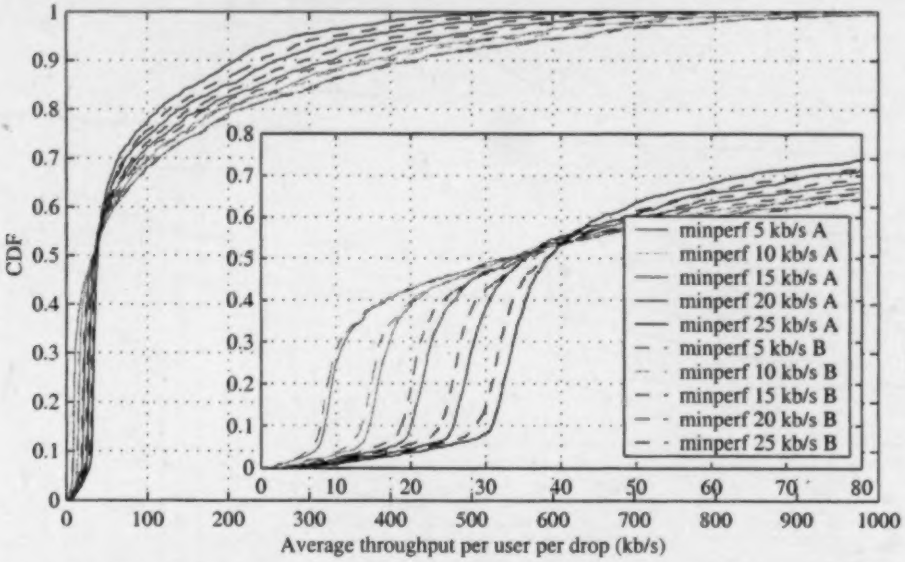


Figure 5.129: Distributions of average throughput per user per drop for the minperf scheduling algorithm in the ITU pedestrian A channel with 16 users per sector using the "PF B" method to calculate the average rate

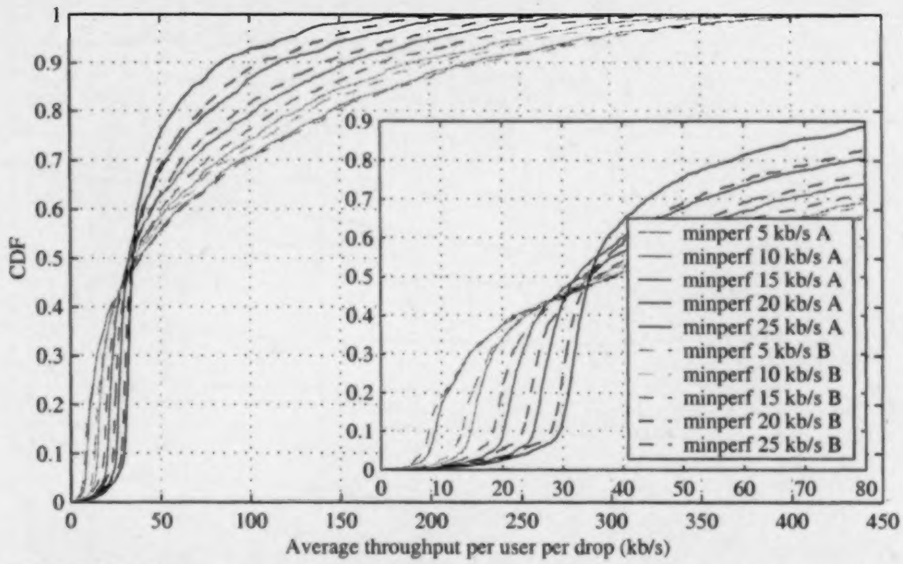


Figure 5.130: Distributions of average throughput per user per drop for the minperf scheduling algorithm in the ITU pedestrian B channel with 16 users per sector using the "PF B" method to calculate the average rate

As the above diagrams show, there is an increased likelihood that a user will have a throughput much higher than the minimum specified. The jump in the distribution occurs closer to the minimum specified throughput using the "PF B" method than is does using the "PF A" method. The key reason for this is that the "PF B" method provides a more accurate calculation of what the actual average rate is than the "PF A" method. Since the latter does not account for early terminations, the value obtained can be lower than the actual throughput. In this event, the minperf algorithm can believe that a user might be receiving a throughput lower than the minimum, even though it may not be. The net effect is that the minperf algorithm tries to boost the throughput of the users with poorer channel conditions when it does not necessarily have to do so. Because the "PF B" calculation method is more accurate, those instances are instead used to transmit to users in much better channel conditions. Since the rates requested by those users are likely to be much higher than the rates requested by those users receiving around the minimum throughput, the sector throughput increases significantly.

The following two tables compare the statistics of the throughput distributions for both calculation methods in the two pedestrian channels. The probabilities of not achieving the minimum throughput are also shown.

TABLE XXXII:
STATISTICS FOR AVERAGE THROUGHPUT PER USER PER DROP DISTRIBUTIONS AND PROBABILITY OF NOT ACHIEVING MINIMUM SPECIFIED THROUGHPUT FOR THE MINIMUM PERFORMANCE SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN A CHANNEL WITH 16 USERS PER SECTOR AND USING THE "PF A" AND "PF B" METHODS TO CALCULATE THE AVERAGE RATE

Calc. Method	"PF A"					"PF B"				
minperf C_k (kb/s)	5	10	15	20	25	5	10	15	20	25

Mean (kb/s)	132.1	119.8	107.0	93.4	81.2	133.7	122.9	112.0	100.9	88.9
Median (kb/s)	32.7	33.0	33.7	35.1	38.1	33.0	33.7	34.6	35.7	38.0
Std. Dev. (kb/s)	205.6	178.1	149.5	119.4	91.6	209.0	184.0	158.9	133.2	106.1
P[Throughput $<C_k$] (%)	2.06	3.25	3.81	4.56	5.38	2.13	3.63	4.69	5.56	6.00

TABLE XXXIII:
STATISTICS FOR AVERAGE THROUGHPUT PER USER PER DROP DISTRIBUTIONS AND PROBABILITY OF NOT ACHIEVING MINIMUM SPECIFIED THROUGHPUT FOR THE MINIMUM PERFORMANCE SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN B CHANNEL WITH 16 USERS PER SECTOR AND USING THE "PF A" AND "PF B" METHODS TO CALCULATE THE AVERAGE RATE

Calc. Method	"PF A"					"PF B"				
minperf C_k (kb/s)	5	10	15	20	25	5	10	15	20	25
Mean (kb/s)	80.4	72.3	62.7	53.8	47.1	82.0	75.4	68.4	60.3	52.6
Median (kb/s)	38.2	35.9	33.5	32.2	34.5	38.9	37.0	35.7	34.0	34.6
Std. Dev. (kb/s)	92.5	77.3	60.3	44.3	30.2	95.1	82.2	68.6	53.7	38.5
P[Throughput $<C_k$] (%)	1.00	1.75	2.63	3.44	4.94	1.56	2.88	4.00	5.06	6.25

The mean, median, and standard deviation of the average throughput per user consistently goes up when switching from the "PF A" method to the "PF B" method. This means that the throughput increases, but so do the differences in the throughputs between users. Unfortunately, the probability of not achieving the minimum throughput also increases. Because of the increased accuracy of the "PF B" method, users operate much closer to the minimum throughput. It is therefore more likely that a user will dip under that minimum if it goes into outage or simply has to wait for a period of time. With the "PF A" method, there is a bit more leeway for a user's throughput to go down without dropping below the minimum. That leeway comes at the expense of the sector throughput.

The next two graphs compare the effective bit rates achieved using the "PF A" and "PF B" calculation methods. Note that only rates from 122.88 kb/s and up are shown, as the proportions for lower rates are negligible.

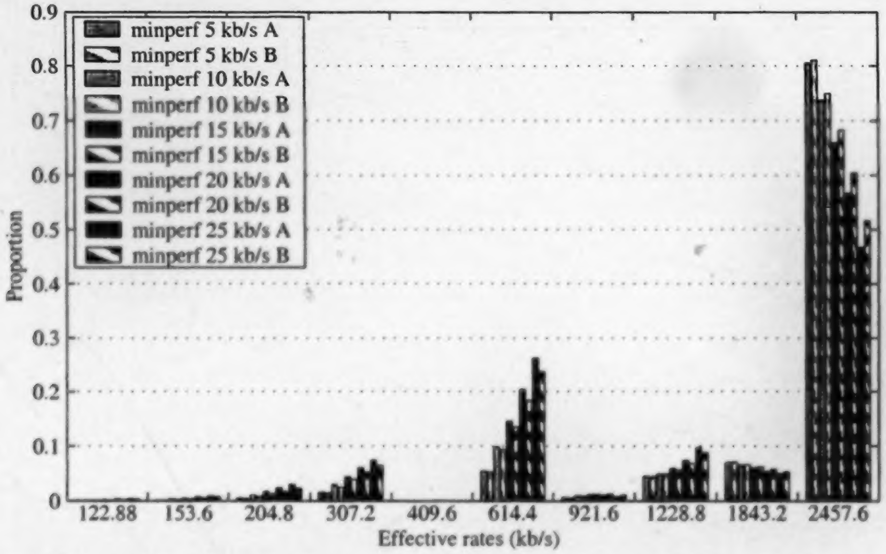


Figure 5.131: Distributions of effective bit rates for packets transmitted under the minperf scheduling algorithm in the ITU pedestrian A channel with 16 users per sector using the "PF B" method to calculate the average rate

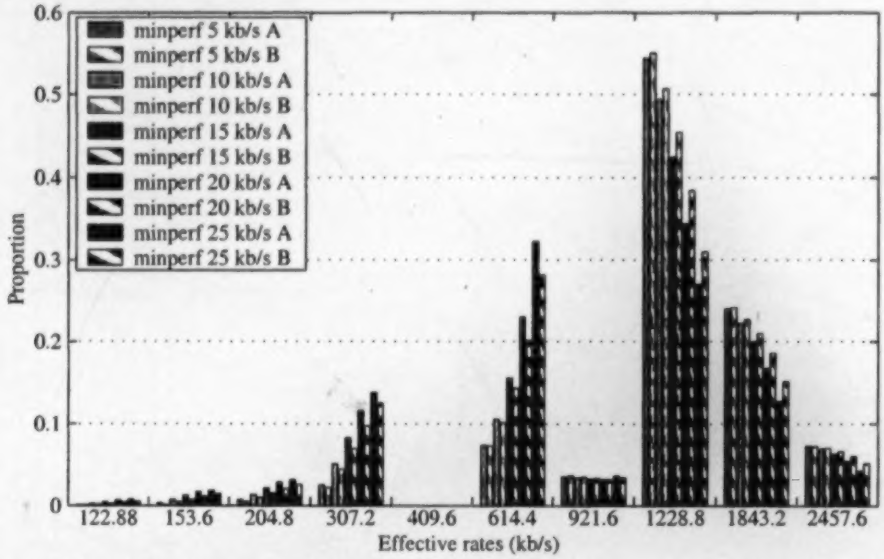


Figure 5.132: Distributions of effective bit rates for packets transmitted under the minperf scheduling algorithm in the ITU pedestrian B channel with 16 users per sector using the "PF B" method to calculate the average rate

The above figures confirm how the throughput increases using the "PF B" method. It can be seen that the proportion of packets sent using the highest rates increases from the "PF A" method to the "PF B" method, while the proportion of packets at lower rates decreases. Thus, fewer lower rate packets are being sent. These would belong to the users receiving around the minimum throughput. Because of the more accurate calculation of the "PF B" method, those users operate closer to that minimum, and as such more packets are sent to the users in good channel conditions at the highest bit rates.

The next two diagrams compare the formats and number of transmissions used to transmit packets under the minperf algorithm using both methods to calculate the average rate.

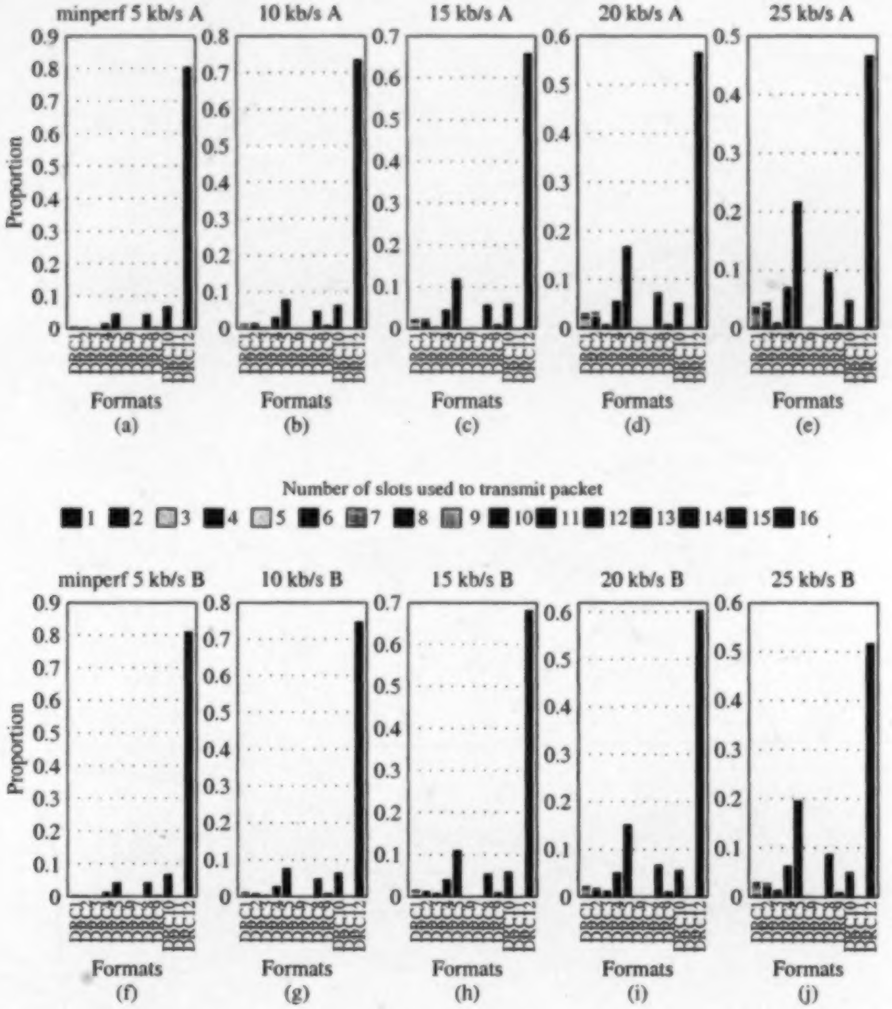


Figure 5.133: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the minperf algorithm in the ITU pedestrian A channel with 16 users per sector using the (a)-(e) "PF A" and (f)-(j) "PF B" methods to calculate the average rate

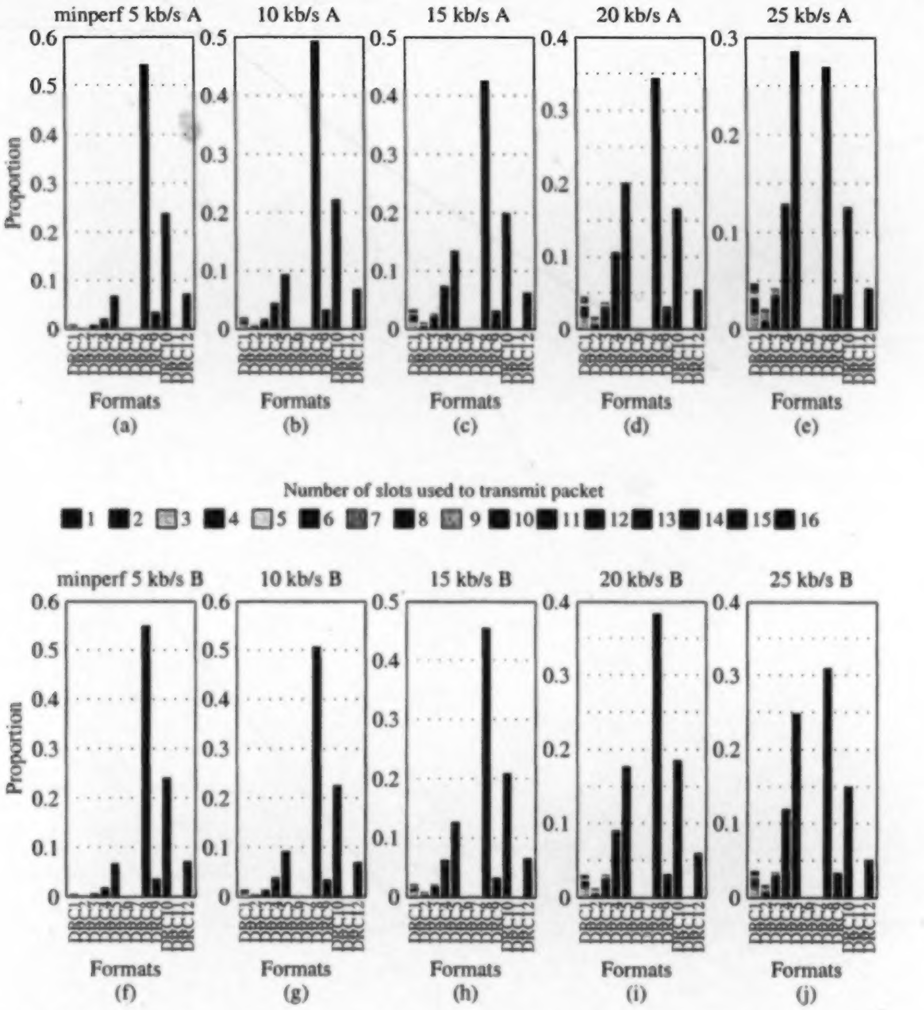


Figure 5.134: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the minperf algorithm in the ITU pedestrian B channel with 16 users per sector using the (a)-(e) "PF A" and (f)-(j) "PF B" methods to calculate the average rate

The distributions of formats show the same results as the distributions of rates; namely, there is a larger proportion of packets sent at the highest rates and a smaller proportion sent with the lower rates.

The following two diagrams show the average delay per packet per drop as a function of the number of users per sector, comparing the "PF A" and "PF B" methods.

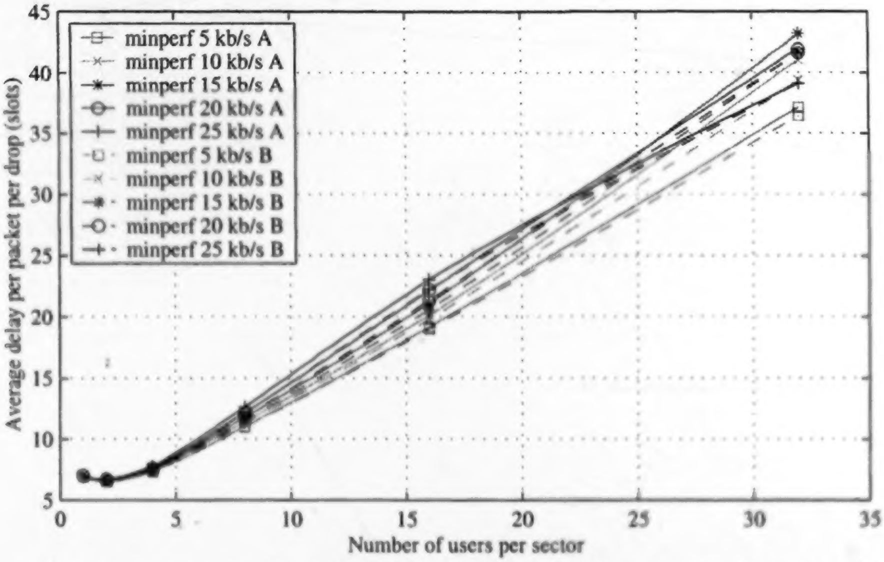


Figure 5.135: Average delay per packet per drop versus number of users per sector for the minperf scheduling algorithm in the ITU pedestrian A channel using the "PF B" method to calculate the average rate

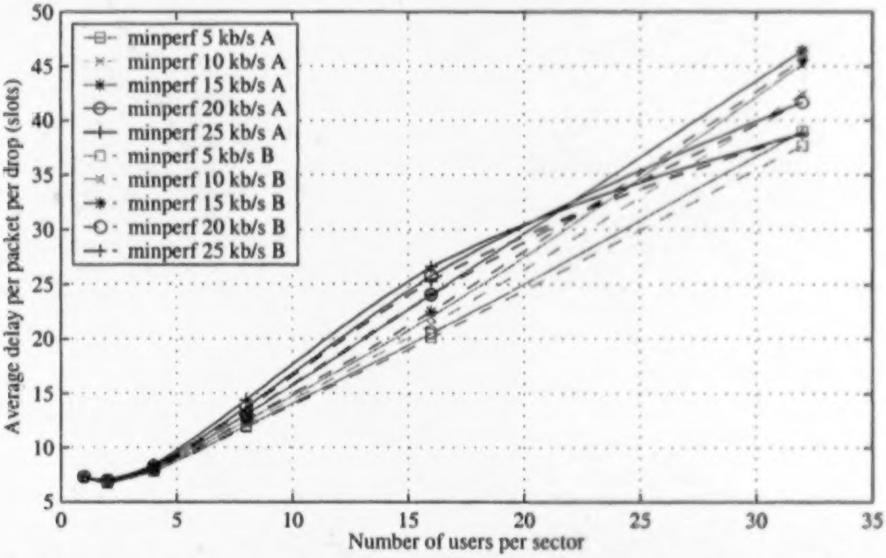


Figure 5.136: Average delay per packet per drop versus number of users per sector for the minperf scheduling algorithm in the ITU pedestrian B channel using the "PF B" method to calculate the average rate

Corresponding to the increased throughput, there is also a decrease in the average delay per packet for the "PF B" method. With more packets sent at higher bit rates, those same packets will also take fewer slots to transmit, leading to a decreased delay overall. This same correspondence between increased throughput and decreased delay per packet has been seen several times in the past for other algorithms.

The next two diagrams show the average packet delay per user per drop as a function of the number of users per sector, comparing the two average rate calculation methods.

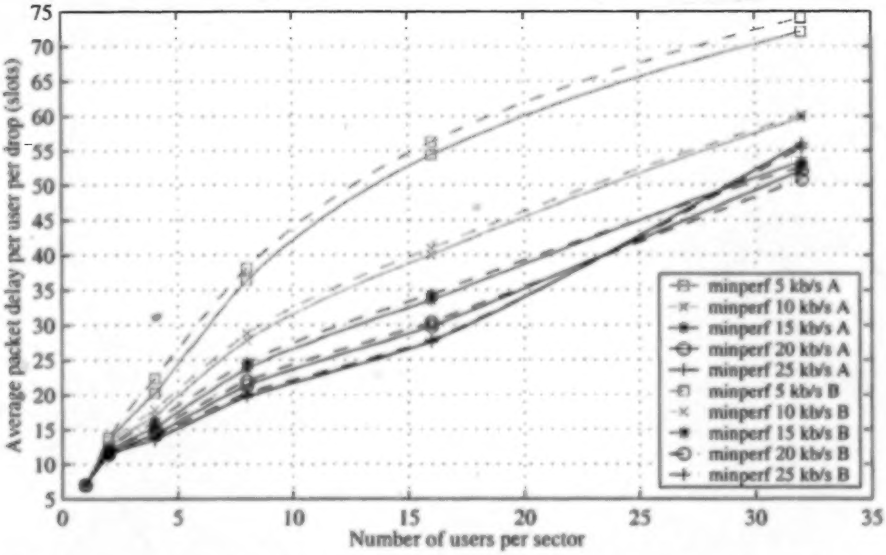


Figure 5.137: Average packet delay per user per drop versus number of users per sector for the minperf scheduling algorithm in the ITU pedestrian A channel using the "PF B" method to calculate the average rate

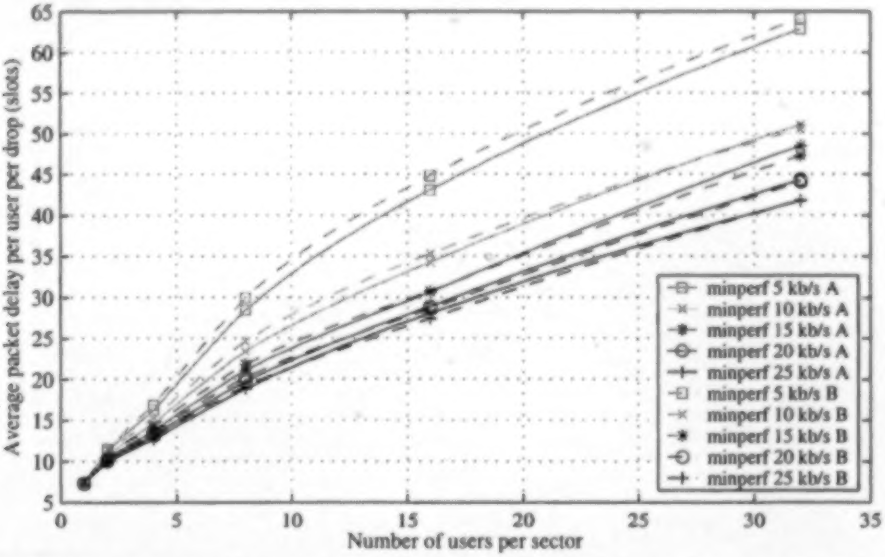


Figure 5.138: Average packet delay per user per drop versus number of users per sector for the minperf scheduling algorithm in the ITU pedestrian B channel using the "PF B" method to calculate the average rate

As seen with previous algorithms, the increase in throughput from the "PF B" method also results in an increase in the average packet delay per user.

However, in absolute terms, the increase really is not too great. The additional amount a user waits on average is at maximum only about 2 slots. Note that in some of the extreme cases (i.e. large numbers of users per sector and a high minimum specified throughput), the delay per user actually decreases. This effect is related to the fact that most users in these cases only receive about the minimum throughput since the channel is close to capacity. In an ordinary case, the "PF B" method causes more packets to be sent to the best users, leading to a decrease in their delay but an increase in the delay of the users in poorer channel conditions. The overall effect is an increase in the average delay per user, since the amount of the delay increases are larger than the amount of the delay decreases. However, in the extreme cases, the better calculation method of "PF B" allows some of the better users to see that they do in fact meet the minimum specified throughput, whereas under the "PF A" method they may appear to be below. This allows the algorithm to provide more service to the poorer users, lowering their delay. Since the delay of the better users does not increase by a significant amount, the overall effect is that the average delay per user decreases.

The following figures show the distributions of delay per packet with 16 users per sector, comparing the two calculation methods.

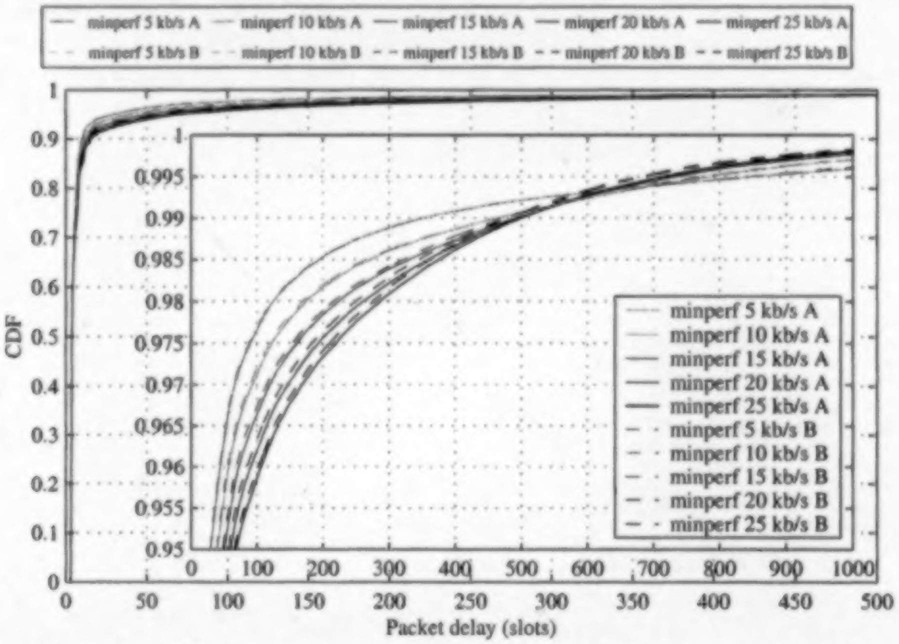


Figure 5.139: Cumulative distribution functions of delays experienced by packets under the minperf scheduling algorithm in the ITU pedestrian A channel with 16 users per sector using the "PF B" method to calculate the average rate

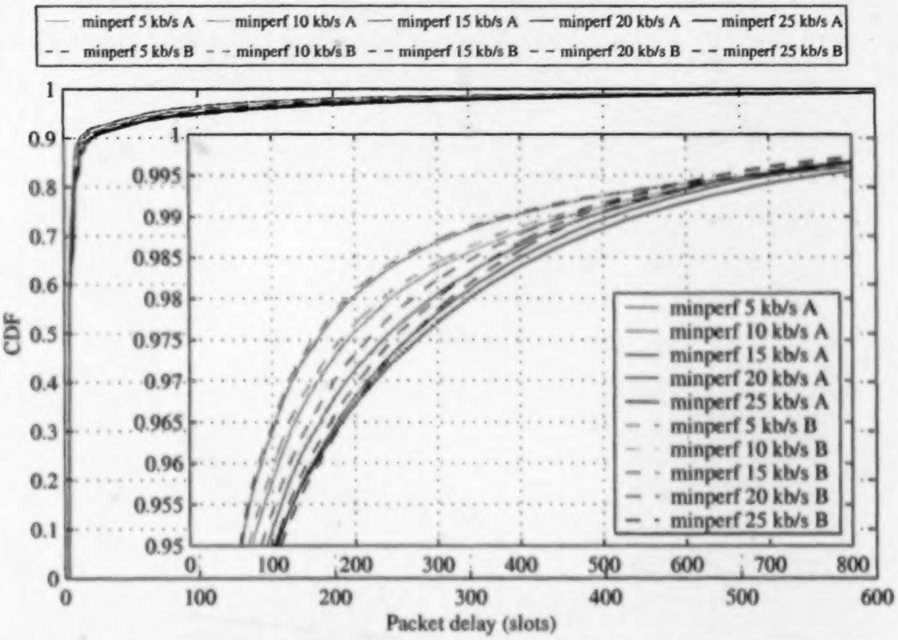


Figure 5.140: Cumulative distribution functions of delays experienced by packets under the minperf scheduling algorithm in the ITU pedestrian B channel with 16 users per sector using the "PF B" method to calculate the average rate

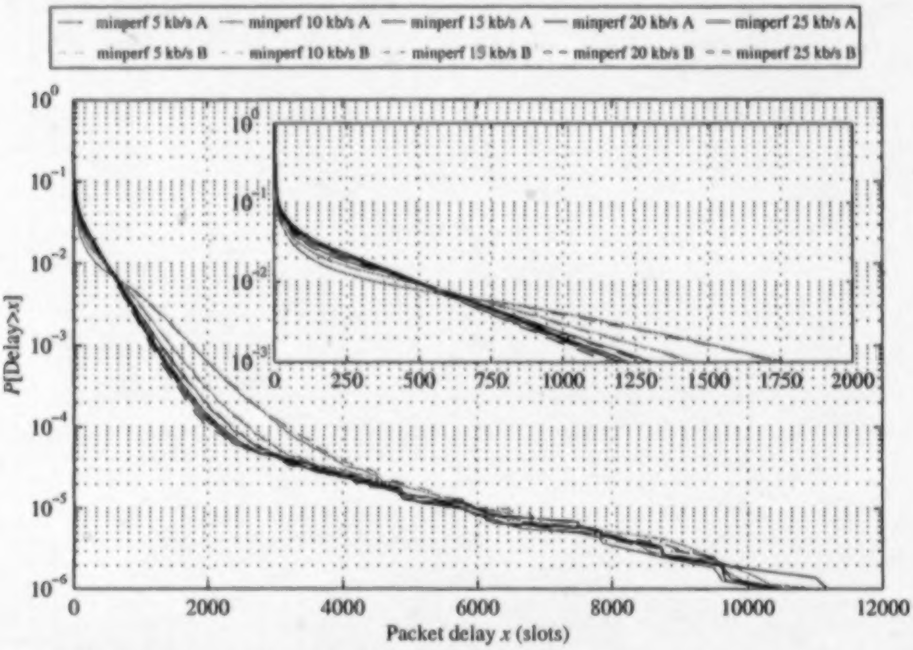


Figure 5.141: Probability of delays experienced by packets being larger than a given value under the minperf scheduling algorithm in the ITU pedestrian A channel with 16 users per sector using the "PF B" method to calculate the average rate

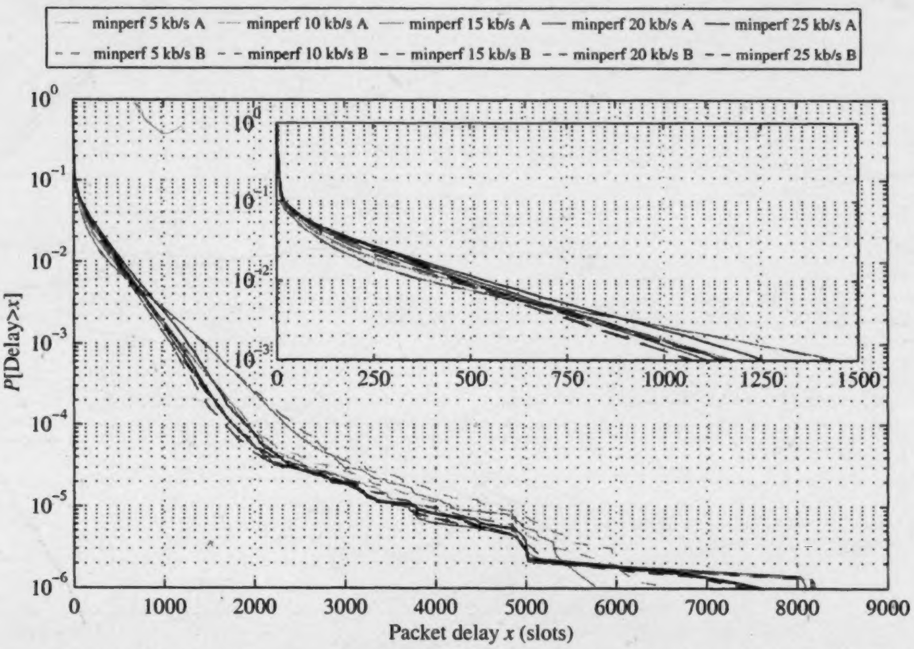


Figure 5.142: Probability of delays experienced by packets being larger than a given value under the minperf scheduling algorithm in the ITU pedestrian B channel with 16 users per sector using the "PF B" method to calculate the average rate

There is a shift in the distributions towards smaller delays for the "PF B" method. However, the shift for the most part does not appear to be very significant. The largest change is in the ITU pedestrian B channel with the higher minimum specified throughputs. The following tables confirm this observation.

TABLE XXXIV:
PERCENTILES AND STANDARD DEVIATIONS OF PACKET DELAYS FOR THE MINIMUM PERFORMANCE SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN A CHANNEL WITH 16 USERS PER SECTOR AND USING THE "PF A" AND "PF B" METHODS TO CALCULATE THE AVERAGE RATE

Calc. Method	"PF A"					"PF B"				
C_k (kb/s) / Percentile	5	10	15	20	25	5	10	15	20	25
50th (median)	4	4	4	4	4	4	4	4	4	4
90th	10.1	11.5	12.5	15.0	15.6	9.9	11.2	11.9	13.1	15.2
95th	29.8	39.9	50.7	60.3	68.4	28.9	38.3	48.1	57.0	65.8
99th	351.1	462.3	490.2	492.9	498.1	352.0	459.8	472.2	476.0	478.0
99.5th	849.8	782.9	733.0	716.3	711.0	843.2	770.2	718.1	688.2	681.5
99.9th	1746.7	1453.1	1307.5	1226.4	1201.1	1756.4	1446.1	1292.5	1198.6	1158.8
99.95th	2129.0	1759.2	1572.1	1483.6	1427.9	2127.1	1751.8	1569.3	1453.3	1387.6
99.99th	3097.6	2610.6	2332.4	2210.9	2115.9	3169.8	2576.7	2350.3	2175.7	2020.5
Std. Dev.	120.51	109.00	103.51	101.98	102.31	120.33	108.14	102.46	99.35	98.52

TABLE XXXV:
PERCENTILES AND STANDARD DEVIATIONS OF PACKET DELAYS FOR THE MINIMUM PERFORMANCE SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN B CHANNEL WITH 16 USERS PER SECTOR AND USING THE "PF A" AND "PF B" METHODS TO CALCULATE THE AVERAGE RATE

Calc. Method	"PF A"					"PF B"				
C_k (kb/s)	5	10	15	20	25	5	10	15	20	25

Percentile										
50th (median)	4	4	4	4	4	4	4	4	4	4
90th	12.0	15.6	19.6	21.7	19.9	11.6	14.1	16.0	19.6	19.9
95th	62.2	75.0	93.2	105.4	103.5	60.3	71.3	85.8	102.1	109.5
99th	389.6	456.7	485.6	511.1	542.1	380.1	442.0	453.9	468.0	484.8
99.5th	691.7	685.1	680.8	701.6	757.8	681.6	670.4	644.0	639.1	670.0
99.9th	1446.1	1183.6	1112.3	1152.9	1257.3	1432.7	1171.4	1072.3	1066.7	1149.9
99.95th	1709.0	1404.3	1288.0	1327.2	1435.5	1719.0	1405.1	1267.4	1227.5	1328.8
99.99th	2335.5	1917.2	1759.0	1749.2	1885.3	2429.8	1945.7	1765.6	1671.3	1759.7
Std. Dev.	100.27	94.03	94.37	98.57	105.47	100.2	92.93	90.24	91.31	96.46

The next two diagrams show the distributions of average packet delay per user with 16 users per sector, comparing the two calculation methods.

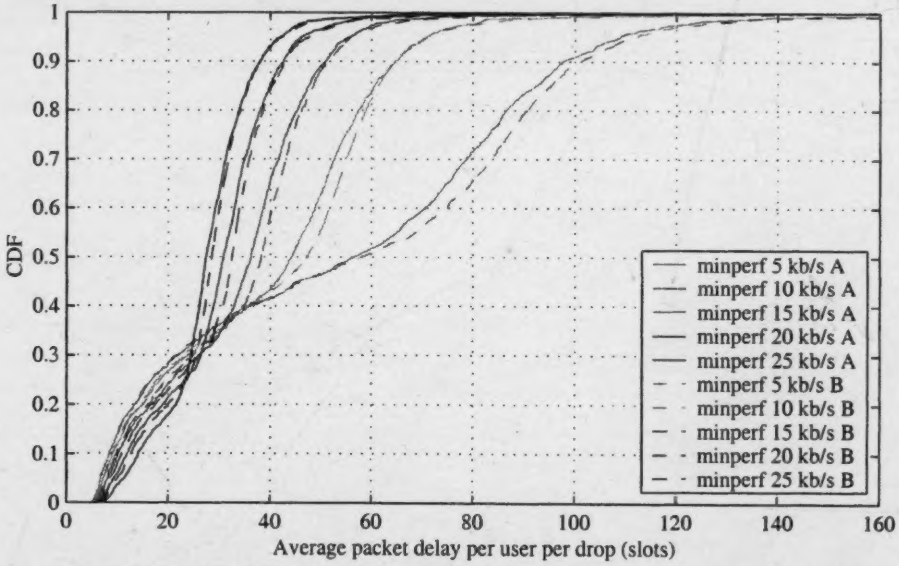


Figure 5.143: Distributions of average packet delay per user for the minperf scheduling algorithm in ITU pedestrian A channel with 16 users per sector using the "PF B" method to calculate the average rates

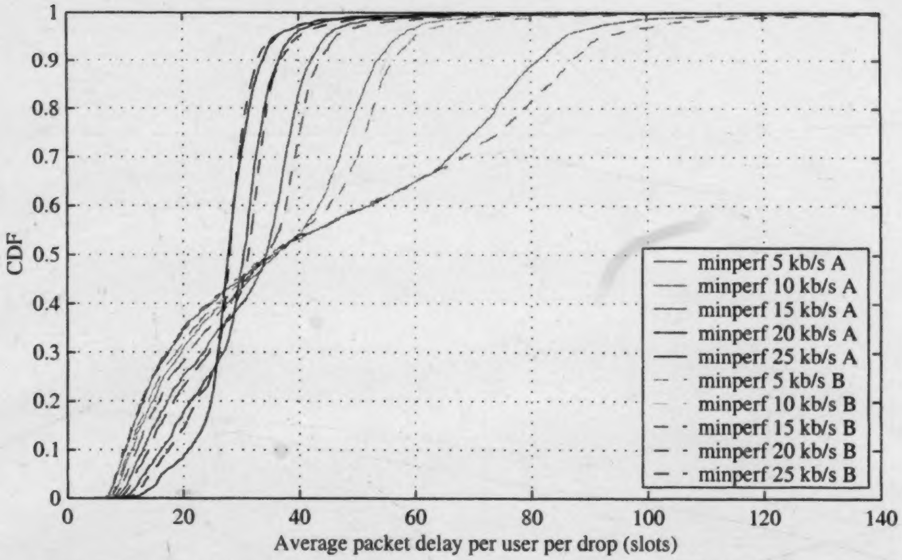


Figure 5.144: Distributions of average packet delay per user for the minperf scheduling algorithm in ITU pedestrian B channel with 16 users per sector using the "PF B" method to calculate the average rates

A shift downwards in delay can be seen at the lower end of the distributions, which represents the higher bit rate users who are served more often under the "PF B" method. Similarly, the delay increases at the upper end of the distribution. This represents those users who receive only the minimum specified throughput and are now served less often. The following table compares the percentiles and standard deviations of the above distributions.

TABLE XXXVI:
PERCENTILES AND STANDARD DEVIATION OF AVERAGE PACKET DELAY PER USER FOR THE MINIMUM PERFORMANCE SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN A CHANNEL WITH 16 USERS PER SECTOR AND USING THE "PF A" AND "PF B" METHODS TO CALCULATE THE AVERAGE RATE

Calc. Method	"PF A"					"PF B"				
C_k (kb/s) / Percentile	5	10	15	20	25	5	10	15	20	25
10th	8.5	9.3	10.3	12.1	14.2	8.5	9.0	9.9	11.0	12.8
25th	16.8	18.6	21.0	23.3	23.4	16.6	18.1	19.7	21.8	23.7
50th (median)	56.7	44.2	36.2	30.8	27.4	58.1	47.1	38.2	32.5	28.5
75th	83.1	54.8	43.1	36.4	32.4	87.1	56.8	44.2	37.1	32.6
90th	97.7	64.8	50.8	42.6	38.0	101.3	64.7	51.3	42.8	37.9
Std. Dev.	38.5	22.8	16.3	12.2	10.1	40.55	23.67	17.28	13.33	10.70

TABLE XXXVII:
PERCENTILES AND STANDARD DEVIATION OF AVERAGE PACKET DELAY PER USER FOR THE MINIMUM PERFORMANCE SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN B CHANNEL WITH 16 USERS PER SECTOR AND USING THE "PF A" AND "PF B" METHODS TO CALCULATE THE AVERAGE RATE

Calc. Method	"PF A"					"PF B"				
C_k (kb/s) / Percentile	5	10	15	20	25	5	10	15	20	25
10th	10.1	11.2	13.0	16.1	20.8	9.8	10.7	12.1	14.1	17.5

25th	14.8	16.6	19.8	24.1	25.4	14.5	16.0	17.8	21.3	24.8
50th (median)	35.7	36.2	34.3	30.4	27.8	34.8	35.5	34.5	31.2	28.0
75th	70.4	48.4	38.6	33.4	30.7	74.7	51.4	40.2	33.9	30.2
90th	81.2	53.5	42.1	36.5	33.8	87.0	56.3	43.7	36.9	32.9
Std. Dev.	31.6	19.5	14.7	10.8	9.11	34.78	22.69	16.07	13.20	11.12

Notice that the standard deviations increase from the "PF A" method to the "PF B" method as the delay each user experiences becomes more varied.

The following two graphs show the distributions of allocated slots per user per drop with 16 users per sector, comparing the two calculation methods.

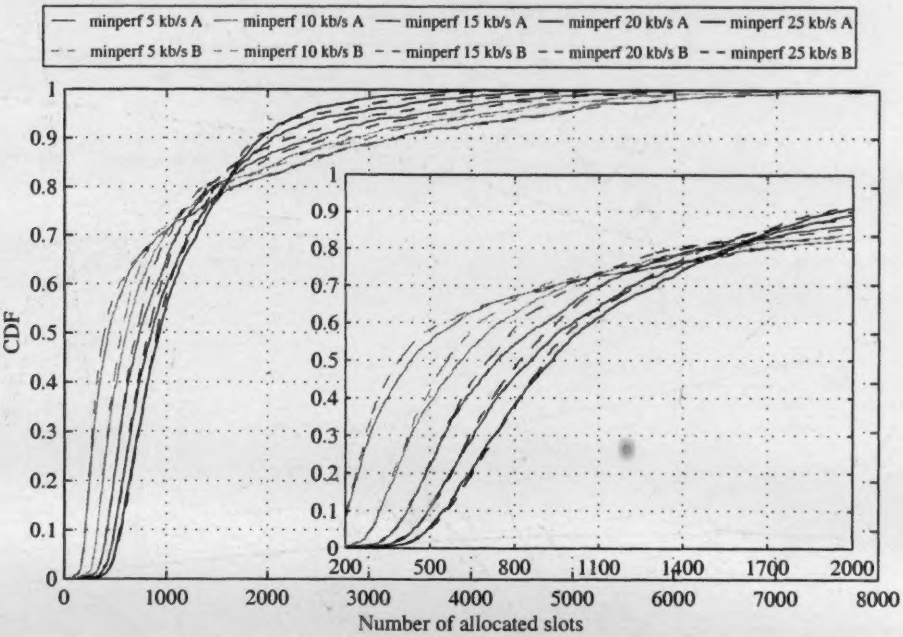


Figure 5.145: Distributions of allocated slots per user per drop for the minperf scheduling algorithm in the ITU pedestrian A channel with 16 users per sector using the "PF B" method to calculate the average rates

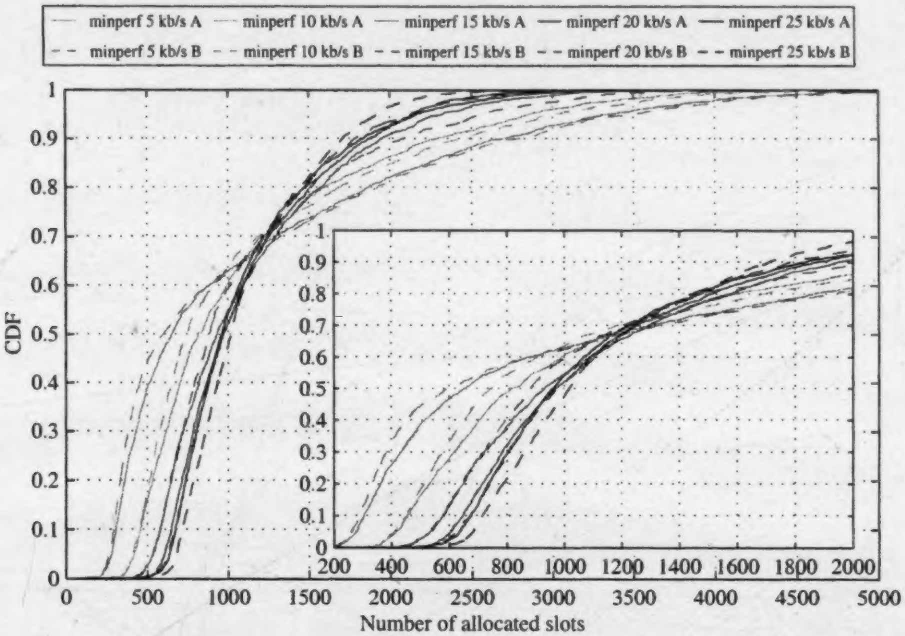


Figure 5.146: Distributions of allocated slots per user per drop for the minperf scheduling algorithm in the ITU pedestrian B channel with 16 users per sector using the "PF B" method to calculate the average rates

The distributions become more varied when the "PF B" method is used. This same effect was seen above for the delay per user distributions, and the reasons for the change are the same as in that prior case. The following tables numerically compare the slot distributions.

TABLE XXXVIII:
PERCENTILES AND STANDARD DEVIATION OF SLOTS ALLOCATED PER USER PER DROP FOR THE MINIMUM PERFORMANCE SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN A CHANNEL WITH 16 USERS PER SECTOR USING THE "PF A" AND "PF B" METHODS TO CALCULATE THE AVERAGE RATE

Calc. Method	"PF A"					"PF B"				
C_k (kb/s) / Percentile	5	10	15	20	25	5	10	15	20	25
10th	210	324	419	493	546	204	324	417	494	561
25th	264	406	519	609	681	257	398	516	610	690
50th (median)	434	609	737	858	933	393	562	705	818	908
75th	1297	1168	1243	1375	1417	1311	1185	1150	1235	1335
90th	3278	2841	2470	2040	1939	3299	3010	2596	2258	1907
Std. Dev.	1499.7	1250.5	1007.8	782.0	621.2	1531.1	1301.3	1079.7	866.9	671.5

TABLE XXXIX:
PERCENTILES AND STANDARD DEVIATION OF SLOTS ALLOCATED PER USER PER DROP FOR THE MINIMUM PERFORMANCE SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN B CHANNEL WITH 16 USERS PER SECTOR USING THE "PF A" AND "PF B" METHODS TO CALCULATE THE AVERAGE RATE

Calc. Method	"PF A"					"PF B"				
C_k (kb/s) / Percentile	5	10	15	20	25	5	10	15	20	25
10th	305	458	578	647	673	291	454	578	675	715
25th	390	574	676	751	769	356	544	684	773	828

50th (median)	654	817	930	968	960	600	746	883	974	1018
75th	1576	1424	1416	1379	1320	1617	1444	1334	1337	1324
90th	2704	2296	1948	1872	1836	2794	2451	2095	1786	1711
Std. Dev.	1028.3	786.6	582.6	497.6	501.5	1075.2	857.5	652.2	477.9	397.8

To summarize, the "PF B" method of calculating the average rates causes a marked increase in the throughput. The cause is due to the increased accuracy of the calculation allowing users in poor channel conditions to operate closer to the minimum specified throughput, which further enables the users in better channel conditions that support higher rates to receive more service. However, operating closer to the minimum throughput also results in an increased probability of not achieving that minimum. Also, as seen with previous algorithms, the increased throughput also corresponds to a decreased average delay per packet, an increased average packet delay per user, and a wider variation in the distributions of throughputs, delays, and slots.

5.3.7. Exponential Rule (EXP1A, EXP2A, EXP1B, & EXP2B)

The exponential rule is similar to the M-LWDF set of algorithms in that it multiplies the weighted requested rate times a delay term. However, instead of multiplying by the head-of-line delay itself, it instead multiplies by an exponential function of the delay:

$$\gamma_k r_k(t)\exp\left(\frac{b_k W_k(t)}{1+\sqrt{\overline{bW}}}\right) \tag{5.7}$$

The terms γ_k, $r_k(t)$, and $W_k(t)$ are the weighting of user k, the requested rate, and the HOL delay, and are the same as for the M-LWDF algorithm. The term inside the square root is the weighted average delay across all N users:

$$\overline{bW} = \tfrac{1}{N}\sum_k b_k W_k(t) \tag{5.8}$$

Because all users are deemed equal for the simulation, the b_k term was set to 1 for every user.

As in the case of the M-LWDF algorithm, the weighting term γ_k can be set to a constant, or made inversely proportional to the average rate each user experiences (i.e. $1/\overline{R}_k(t)$). Therefore, the algorithm is also a modification of the maxD algorithm in the first case and the PF algorithm in the second case. Thus, there can be a set of EXP1 and EXP2 simulations corresponding to either case, respectively. However, there is an added twist to this algorithm. When multiplying directly by the HOL delay in the M-LWDF algorithm, it does not matter if the delay is in terms of seconds or in terms of slots; the metric would change by a factor of 1 slot/1.6667×10^{-3} seconds for all users, so which user has the relatively larger metric does not change. However, with the exponential term and the division by the weighted average delay involved within the EXP algorithm, this factor could now affect which users are selected for transmission.

There is therefore a further division of simulations: an "A" set where the delay is in terms of slots, and a "B" set where the delay is in units of seconds. Therefore, there are four variations in total to be simulated: EXP1A (maxD/slots), EXP2A (PF/slots), EXP1B (maxD/seconds), and EXP2B (PF/seconds). Because the metric is again dividing by the average rate and not subtracting from it as in the case of the minperf algorithm, only the "PF A" method was used to calculate the average rate for the EXP2 simulations.

The following two graphs show the average sector throughput per drop as a function of the number of users per sector for the EXP algorithm.

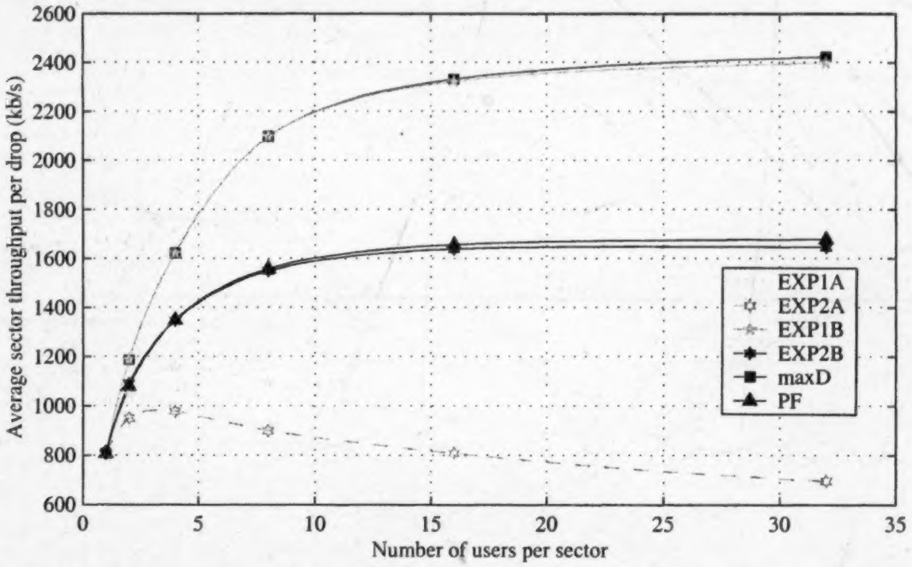


Figure 5.147: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian A channel for the EXP scheduling algorithm

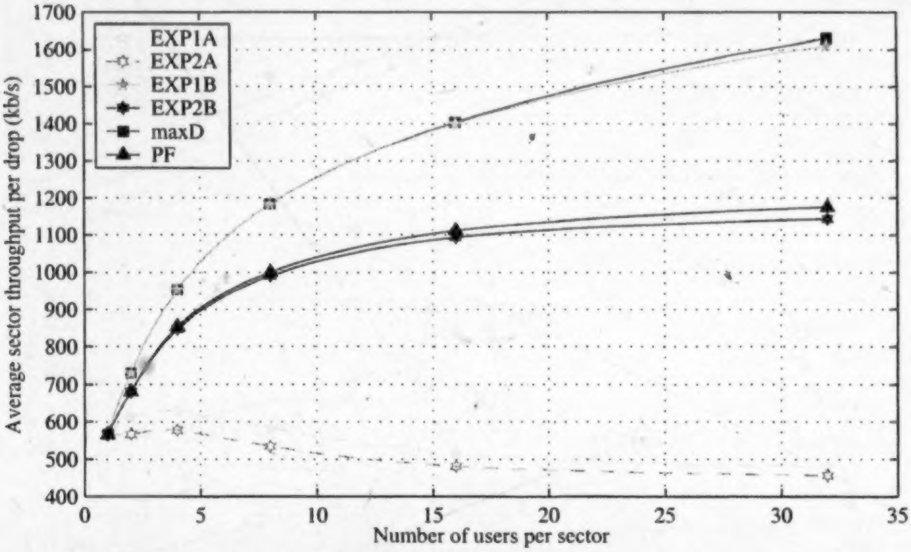


Figure 5.148: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian B channel for the EXP scheduling algorithm

The throughput results of the EXP1A and EXP2A algorithms are extremely poor. The reason for this is that using slots for the units of delay creates an extreme and undue emphasis on the delay, which almost constantly overrides the channel conditions. Under units of slots, only a few slots worth of delay are required to quickly raise the value of the average delay to be much larger than the 1 in the denominator of the exponential term. The whole exponential term then grows extremely rapidly, and becomes the dominant factor in deciding which user should receive a transmission. The algorithm for the most part simply transmits to the user who has experienced the largest delay, almost regardless of the channel conditions. The requested rate still does play a small role, and consequently the throughput is above that achieved by the RR algotihm, although not by much.

Comparatively, the EXP1B and EXP2B algorithms show a very good throughput response. In fact, the average throughput is only slightly below that achieved by the maxD and the PF algorithms, respectively. In these two cases, the delay plays a much less significant role. The exponential term does not really come into play until the HOL delay of a user is on the order of 1 second, such that the numerator is about the same order of magnitude as the denominator in the exponential term. At that point, the delay becomes increasingly important, and the user with the long delay is more and more likely to be served. Until the delay builds up for a user, however, the algorithm operates as if it were the maxD algorithm or the PF algorithm, respectively.

The next two diagrams show the distributions of throughput per user per drop with 16 users per sector for the EXP algorithm.

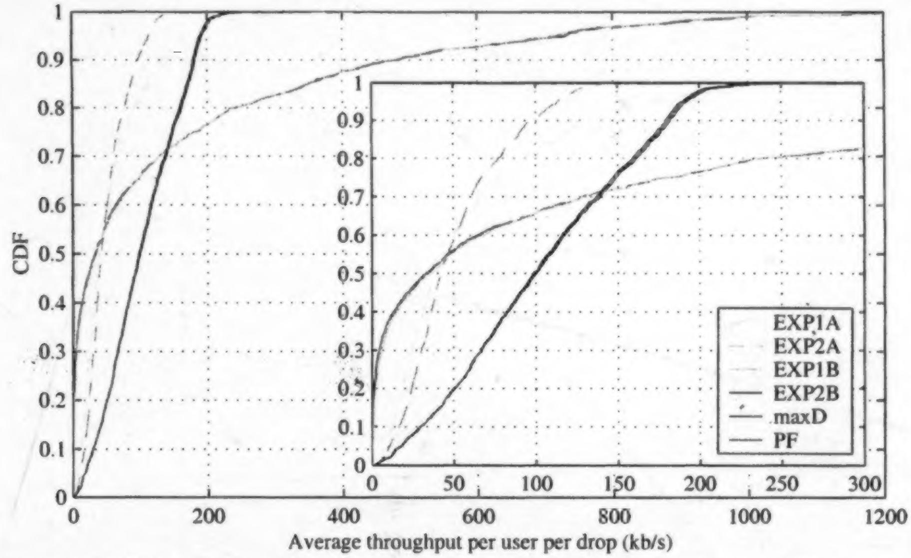


Figure 5.149: Distributions of average throughput per user per drop for the EXP scheduling algorithm in the ITU pedestrian A channel with 16 users per sector

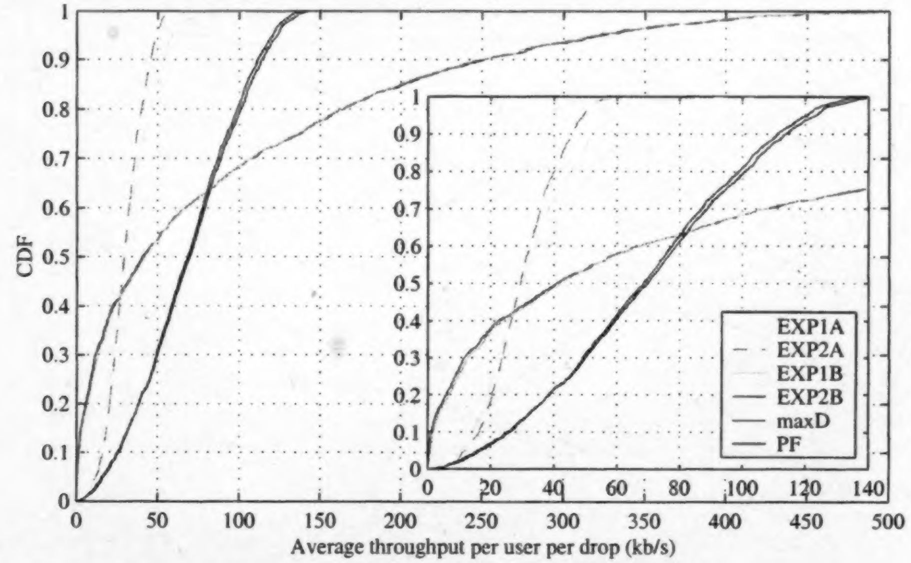


Figure 5.150: Distributions of average throughput per user per drop for the EXP scheduling algorithm in the ITU pedestrian B channel with 16 users per sector

The throughputs seen by users under the EXP1A and EXP2A algorithms are quite small, again due to the exponential term overriding the channel conditions. For the EXP1B and EXP2B algorithms, the distributions are almost identical to the maxD and PF distributions, respectively. The following table compares some statistics of the above distributions.

TABLE XL:
STATISTICS FOR AVERAGE THROUGHPUT PER USER PER DROP DISTRIBUTIONS FOR THE EXPONENTIAL RULE SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN A AND B CHANNELS WITH 16 USERS PER SECTOR

Channel	ITU pedestrian A						ITU pedestrian B					
Algorithm	EXP1A	EXP2A	EXP1B	EXP2B	maxD	PF	EXP1A	EXP2A	EXP1B	EXP2B	maxD	PF
Mean (kb/s)	58.9	50.5	145.1	102.5	145.7	103.6	32.2	30.1	87.5	68.4	87.7	69.5
Median (kb/s)	39.3	42.6	35.2	98.5	33.4	99.6	30.1	29.4	41.9	68.5	41.0	69.6
Std. Dev. (kb/s)	55.3	31.0	232.9	54.7	236.8	55.6	15.3	11.1	105.0	31.5	106.5	32.5

The next two diagrams show the distributions of effective bit rates with 16 users per sector.

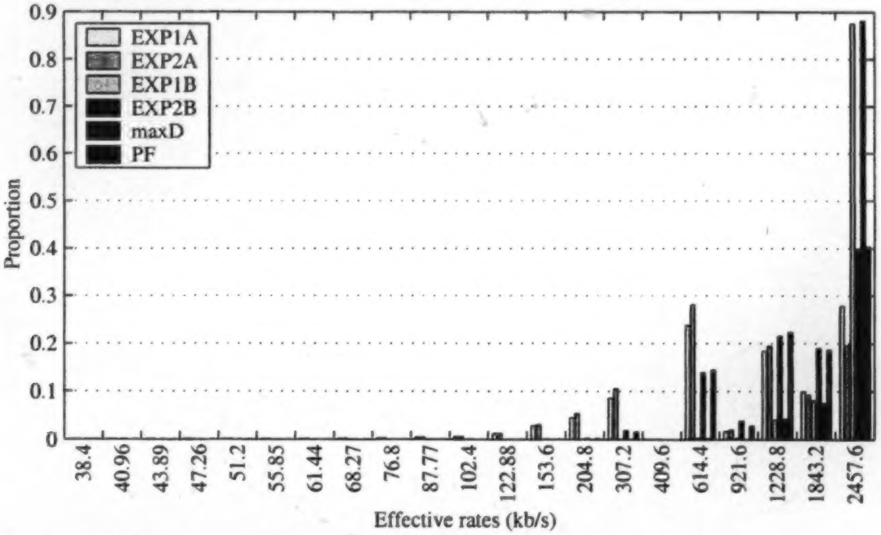


Figure 5.151: Distributions of effective bit rates for packets transmitted under the EXP scheduling algorithm in the ITU pedestrian A channel with 16 users per sector

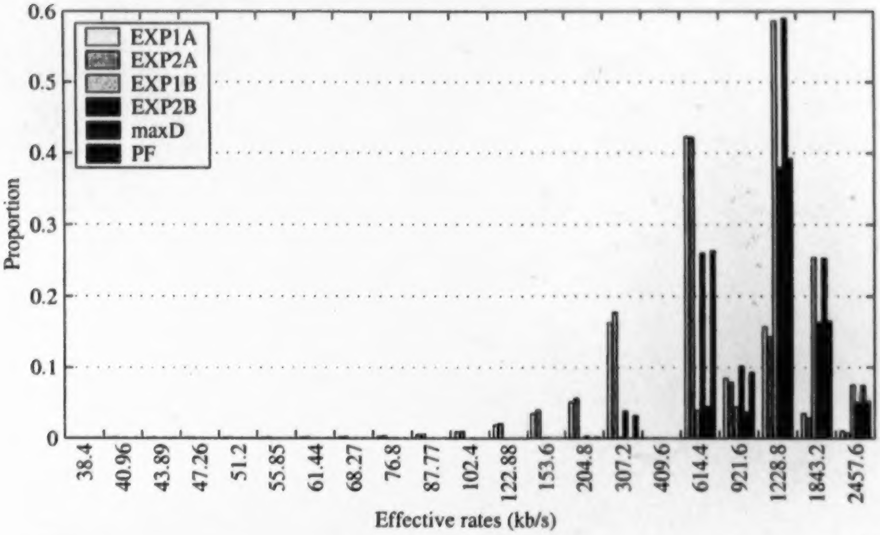


Figure 5.152: Distributions of effective bit rates for packets transmitted under the EXP scheduling algorithm in the ITU pedestrian B channel with 16 users per sector

The following two diagrams show the distributions of formats and number of transmissions used to send packets under the EXP algorithm with 16 users per sector.

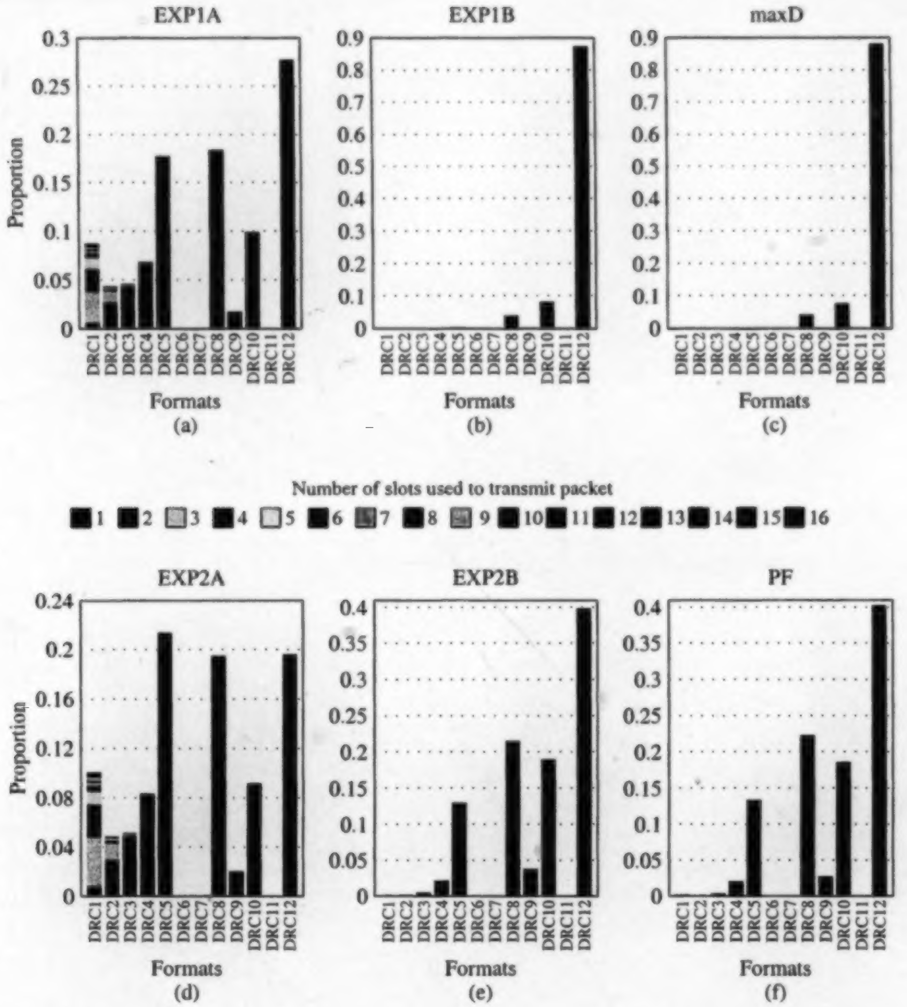


Figure 5.153: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a) EXP1A, (b) EXP1B, (c) maxD, (d), EXP2A, (e) EXP2B, and (f) PF scheduling algorithms in the ITU pedestrian A channel with 16 users per sector

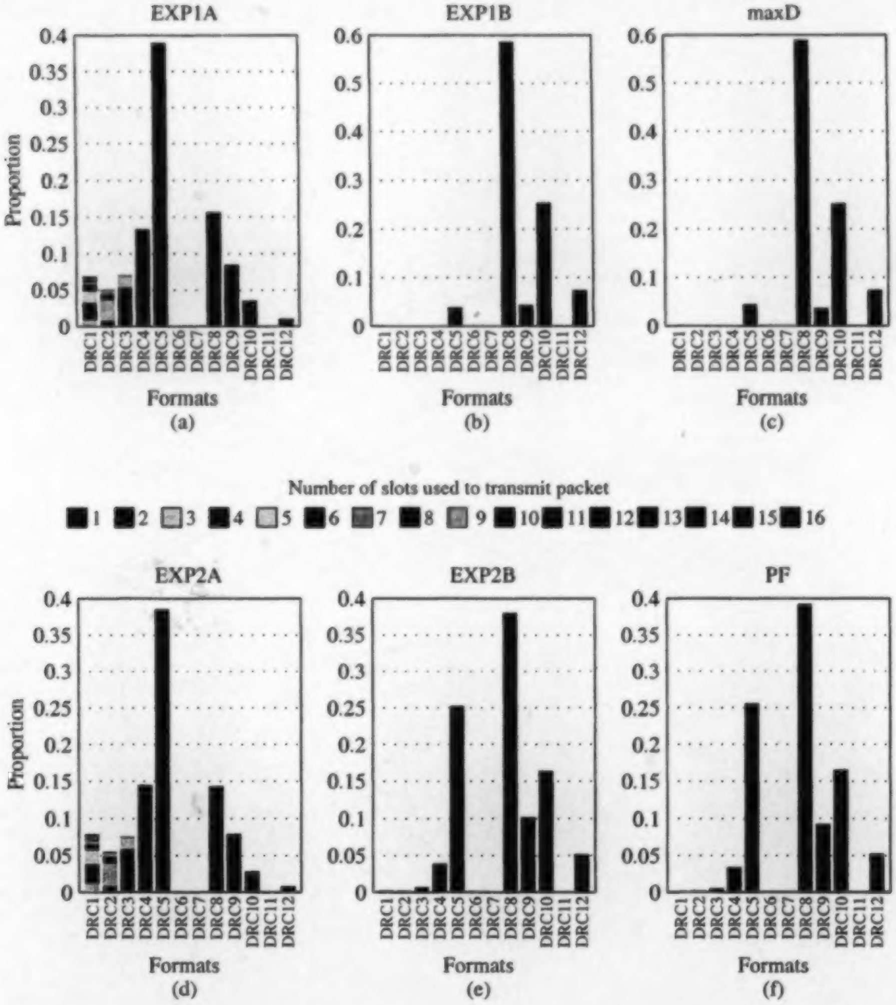


Figure 5.154: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a) EXP1A, (b) EXP1B, (c) maxD, (d), EXP2A, (e) EXP2B, and (f) PF scheduling algorithms in the ITU pedestrian B channel with 16 users per sector

There is a slightly lower probability of the transmission at the highest rates and a slightly higher probability of lower rates for the "B" algorithms compared to their non-EXP counterparts. This corresponds to the lower rate users being compensated for long delays. However, the difference is very minor, and there again are no significant differences in the distributions. Compare this to the "A" algorithms, where there is a much higher probability of lower rate transmissions, since the channel conditions are almost constantly being overridden.

The following figures show the average delay per packet per drop as a function of the number of users per sector for the EXP algorithm.

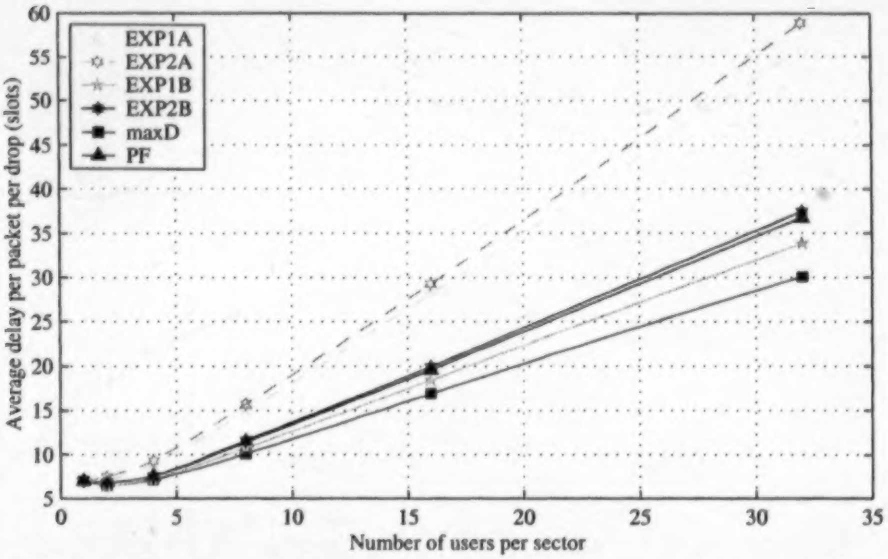


Figure 5.155: Average delay per packet per drop versus number of users per sector for the EXP scheduling algorithm in the ITU pedestrian A channel

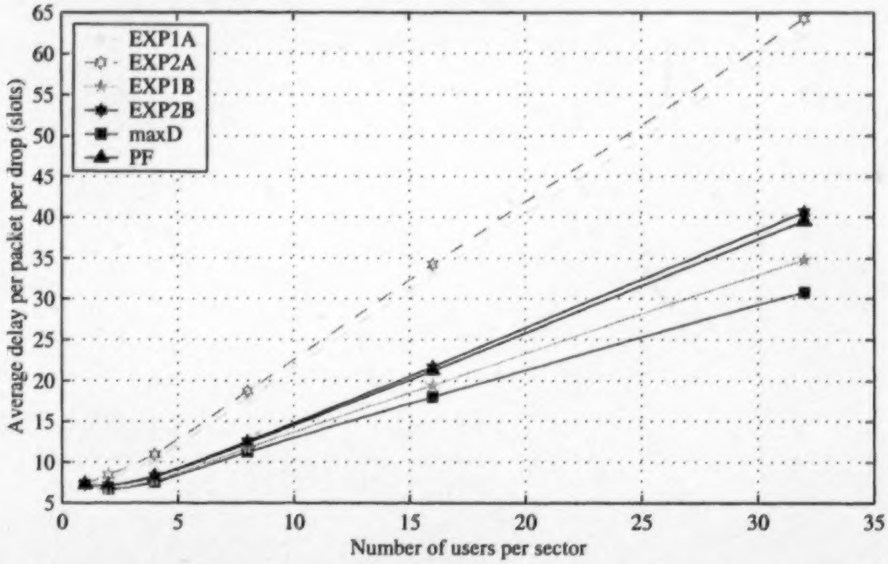


Figure 5.156: Average delay per packet per drop versus number of users per sector for the EXP scheduling algorithm in the ITU pedestrian B channel

As expected after seeing the notably low throughput from the EXP1A and EXP2A algorithms, the average delay per packet for those two algorithms is very high. The delay is not quite as high as for the RR algorithm, which shows that channel conditions do still affect the scheduling decision, albeit in a minor fashion. Examining the EXP1B and EXP2B algorithms, the delays seen in those two cases are clearly much better. There is an increase in delay from the PF algorithm to the EXP2B algorithm, but the change is minor. More significant is the increase in delay from the maxD algorithm to the EXP1B algorithm as the number of users per sector increases. Note that the delay increases a fair amount, even though the throughput does not decrease anywhere near as much. This essentially is due to the fact that the maxD algorithm always chooses the user with the absolute highest requested rate. Therefore, even though another user may request a rate half of the highest, for example, that user will not be selected, even though the requested rate is still quite high. Hence, the delay for that user will build up until its exponential term in the metric becomes significant, at which point the user would be scheduled. Since the requested rate is not that much smaller than the maximum, the average throughput does not decrease very much. However, because the delays for that user and others in a similar situation are quite high, the average delay per packet increases by a notable amount. The question becomes one of frequency. Under the maxD algorithm, there are a few packets with extremely large delays. Under the EXP1B algorithm, the delays are relatively smaller, but there are more of them that are transmitted with that delay. Hence, the average delay per packet increases.

The following two diagrams show the average packet delay per user as a function of the number of users per sector for the EXP algorithm.

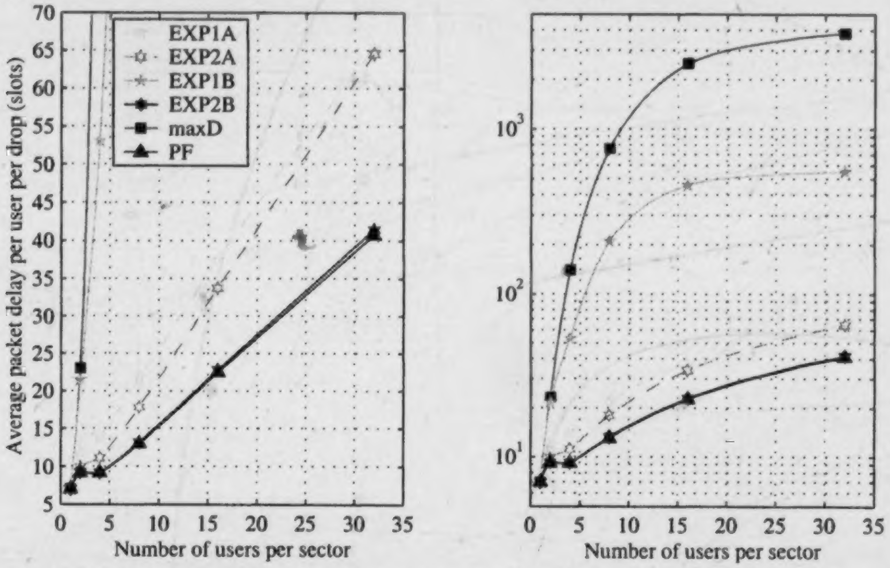


Figure 5.157: Distributions of average packet delay per user for the EXP scheduling algorithm in ITU pedestrian A channel with 16 users per sector

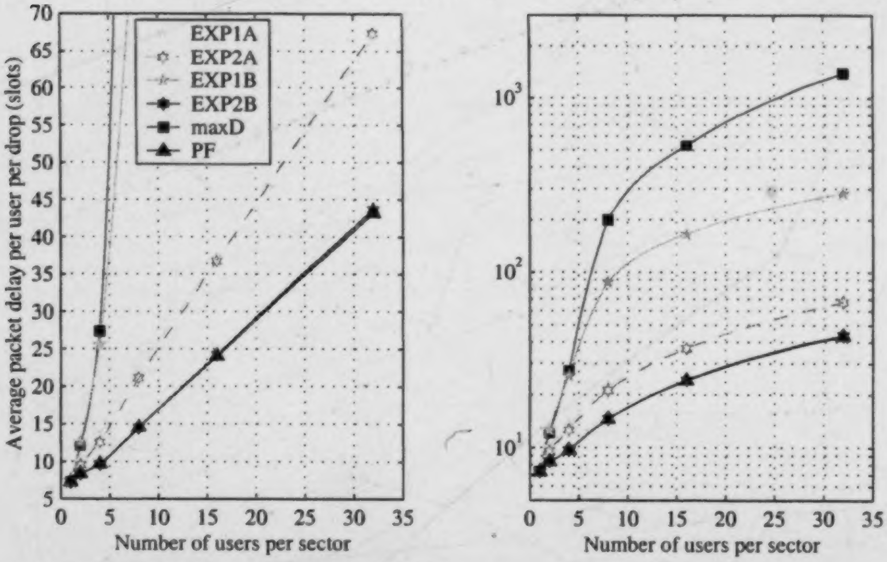


Figure 5.158: Distributions of average packet delay per user for the EXP scheduling algorithm in ITU pedestrian B channel with 16 users per sector

The average packet delay per user for the EXP1B algorithm is much lower than that for the maxD algorithm. The added exponential term does not allow users to be starved for service for extended periods of time, assuming that the users can support any non-zero rate at all. Hence, without that possibility of extended delay, the average delay per user naturally will decrease by a significant amount. In fact, the average delay is becomes about an order of magnitude lower under the EXP1B algorithm than under the maxD algorithm. There is no significant change between the PF and the EXP2B algorithms, as there is no predispositioned tendency towards long delays in the first place. Not surprisingly, the delays seen under the EXP1A and EXP2A algorithms are on the order of those seen under the RR algorithm.

The following diagrams show the distributions of delay per packet with 16 users per sector for the EXP algorithm.

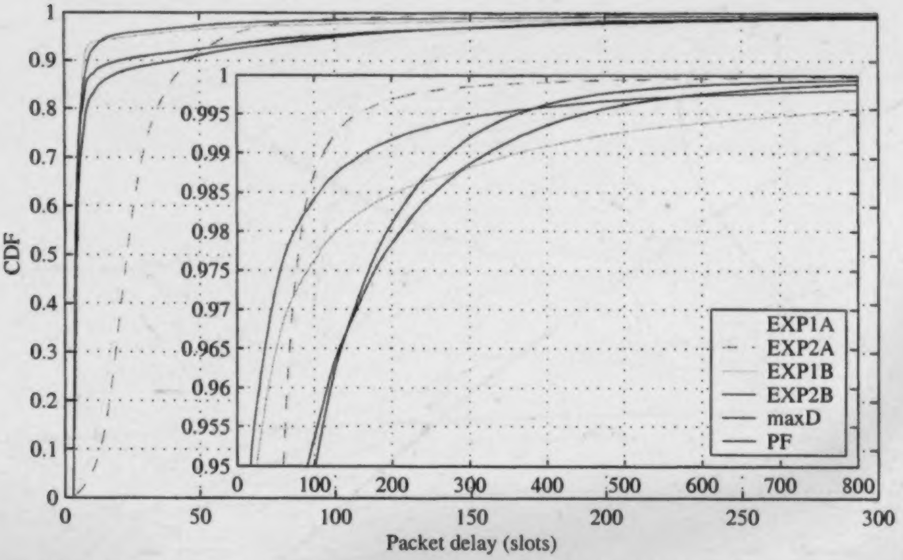


Figure 5.159: Cumulative distribution functions of delays experienced by packets under the EXP scheduling algorithm in the ITU pedestrian A channel with 16 users per sector

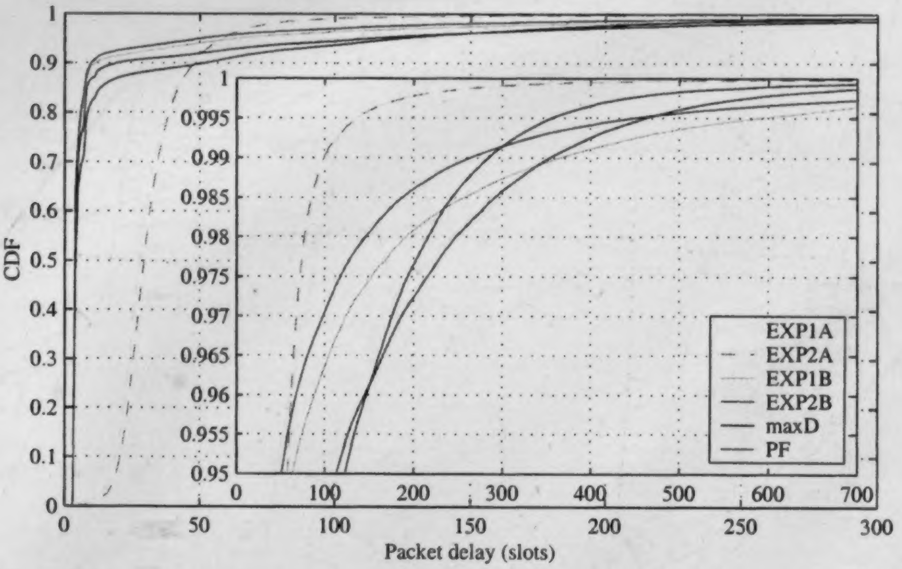


Figure 5.160: Cumulative distribution functions of delays experienced by packets under the EXP scheduling algorithm in the ITU pedestrian B channel with 16 users per sector

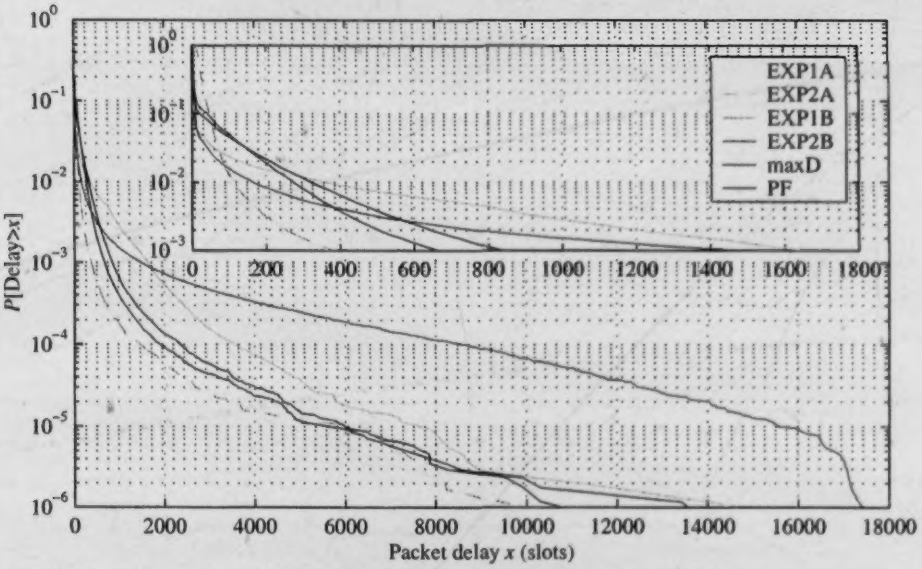


Figure 5.161: Probability of delays experienced by packets being larger than a given value under the EXP scheduling algorithm in the ITU pedestrian A channel with 16 users per sector

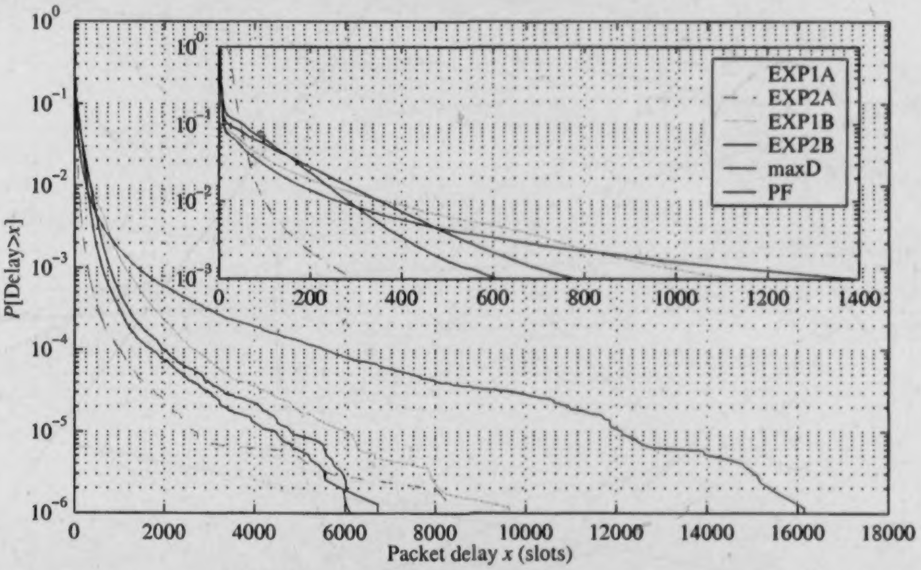


Figure 5.162: Probability of delays experienced by packets being larger than a given value under the EXP scheduling algorithm in the ITU pedestrian B channel with 16 users per sector

The probability of small packet delays is somewhat lower under the EXP1B algorithm than the maxD algorithm, since service is taken away from the highest rate users to compensate for the delay of the other users. This results in more packets with intermediate delays instead of a few packets with long delays. It can also be seen that the EXP1B algorithm displays a much smaller probability of extremely long delays (e.g. larger than 2000 slots). For even larger delays, the probability of a packet seeing that delay is at least an order of magnitude lower under the EXP1B algorithm than under the maxD algorithm. Therein lies one of the key strengths of the EXP algorithm: it reduces the probability of a packet or a user experiencing an extremely large delay with very little sacrifice in throughput. For the EXP2B algorithm, there is a general reduction in the probability of a given delay for almost the entire range of delay values. The delays seen under the EXP1A and EXP2A are very similar to each other and smaller than the other algorithms, but that reduced delay comes with a major sacrifice in throughput, as already seen. The following table compares the percentiles and standard deviations of the above distributions.

TABLE XLI:
PERCENTILES AND STANDARD DEVIATIONS OF PACKET DELAYS FOR THE EXPONENTIAL RULE SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN A AND B CHANNELS WITH 16 USERS PER SECTOR

Channel	ITU pedestrian A						ITU pedestrian B					
Algorithm / Percentile	EXP1A	EXP2A	EXP1B	EXP2B	maxD	PF	EXP1A	EXP2A	EXP1B	EXP2B	maxD	PF
50th (median)	21.5	22.9	4.8	4.2	4	4	27.7	29.3	4.2	4	4	4
90th	47.7	46.4	7.6	40.9	8.9	20.9	47.6	47.6	11.7	51.5	9.9	19.6
95th	62.6	60.0	25.3	100.7	17.3	90.4	59.4	58.2	64.0	122.9	51.1	113.0
99th	115.7	112.6	362.1	274.6	164.7	322.6	101.0	99.1	365.7	283.4	263.2	353.6
99.5th	156.9	157.5	735.0	361.5	329.3	448.3	137.1	136.5	590.6	356.1	449.6	461.3
99.9th	364.8	382.7	1632.6	666.3	1476.4	837.9	294.7	297.0	1127.6	609.4	1399.0	777.2
99.95th	531.9	552.2	2047.3	894.4	2835.4	1095.7	411.1	422.1	1502.2	820.7	2245.7	998.5
99.99th	1341.4	1471.6	3571.8	1909.3	8673.6	2345.1	958.6	1014.2	2650.7	1709.0	5549.0	1987.0
Std. Dev.	42.34	44.62	116.60	69.71	168.80	80.64	31.37	31.86	92.25	65.93	132.28	76.93

Of note is the fact that the EXP algorithm in most cases raises the median packet delay above 4 slots, meaning that less than half the packets under that algorithm are sent in groups to the same user. The proportion is still close to half for the "B" cases, though, since the median is only slightly above 4 slots.

The following two diagrams show the distributions of average packet delay per user per drop with 16 users per sector for the EXP algorithm.

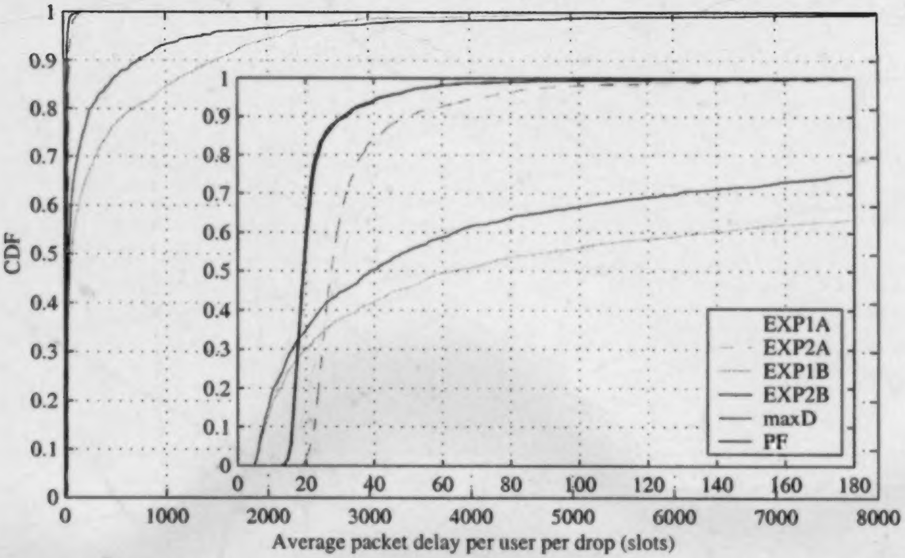


Figure 5.163: Distributions of average packet delay per user for the EXP scheduling algorithm in ITU pedestrian A channel with 16 users per sector for those users who receive at least one packet during a drop

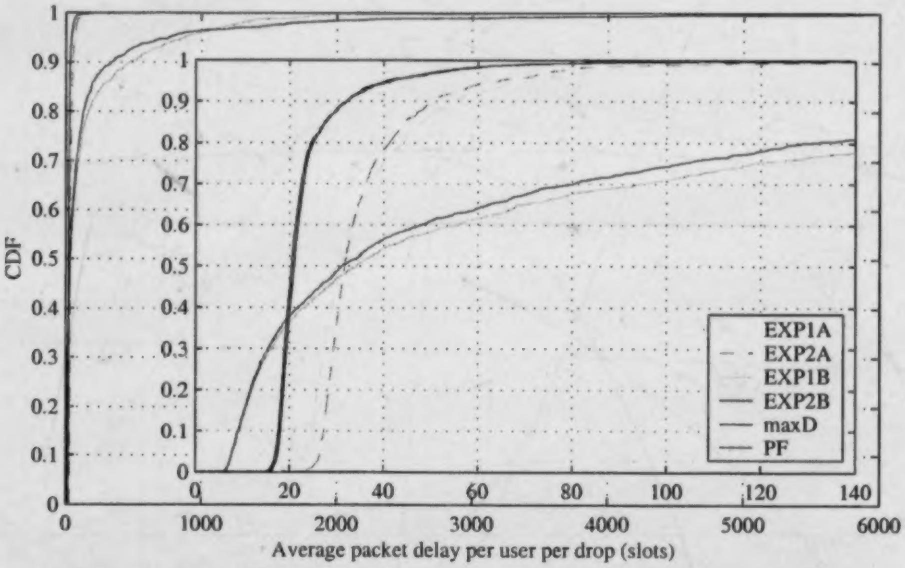


Figure 5.164: Distributions of average packet delay per user for the EXP scheduling algorithm in ITU pedestrian B channel with 16 users per sector for those users who receive at least one packet during a drop

The distributions of delay per user are a bit misleading if not interpreted correctly. It would appear that the maxD algorithm has a somewhat higher probability of users with small delays than the EXP1B algorithm. This is not really true. Because it is impossible to tell how long a packet will wait to be transmitted beyond the end of a simulation drop, those packets which are in the process of being sent or are waiting to be sent when a simulation drop ends are not included in the above distributions. This includes those packets belonging to users starved for service for the entire drop. Note that those starved users are filtered out of the maxD algorithm distribution. However, those same users are no longer starved under the EXP1B algorithm. Instead, the packets of those users will have delays in the low thousands of slots. Because of the filtering process, the delays seen under the EXP1B algorithm can seem like they are longer than under the maxD algorithm. To compare the two distributions more directly, the maxD distribution should be shifted downwards by the probability of a user being starved for service (i.e. about 12% in the ITU pedestrian A channel and 2% in the ITU pedestrian B channel). If this shift was done, the distributions would be very similar up to about 500 slots. Beyond that point, the EXP1B algorithm would show a smaller probability of longer delays.

The PF and EXP2B distributions are also very similar, and display the smallest delays per user. The distributions of the EXP1A and EXP2A algorithms are located at delays higher than the PF or EXP2B algorithms, but below that of the maxD and EXP1B algorithms. The following table compares the percentiles and standard deviations as shown in the graphs above.

TABLE XLII:
PERCENTILES AND STANDARD DEVIATION OF AVERAGE PACKET DELAY PER USER FOR THE EXPONENTIAL RULE SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN A AND B CHANNELS WITH 16 USERS PER SECTOR

Channel	ITU pedestrian A						ITU pedestrian B					
Algorithm / Percentile	EXP1A	EXP2A	EXP1B	EXP2B	maxD	PF	EXP1A	EXP2A	EXP1B	EXP2B	maxD	PF
10th	18.3	22.4	8.0	16.1	7.9	15.9	24.6	27.3	9.3	18.1	9.1	17.7
25th	22.2	24.2	15.4	17.3	13.4	17.0	26.6	28.8	13.8	19.0	13.4	18.7
50th (median)	29.3	27.4	62.9	19.4	39.3	19.1	31.1	31.7	34.3	20.9	31.3	20.4
75th	40.7	34.0	437.3	22.3	186.4	21.8	39.9	38.0	118.3	23.6	102.7	23.2
90th	63.4	50.1	1482.0	31.7	725.9	30.9	54.8	49.9	472.2	32.0	337.9	32.0
Std. Dev.	28.7	21.4	1070.1	13.9	1102.1	14.1	19.0	16.1	375.2	18.5	710.5	21.9

Note that despite the shift between the EXP1B algorithm and the maxD algorithm, the EXP1B algorithm has a smaller standard deviation, primarily because the tail at the upper end of the distribution does not extend to as large values of delay.

The following graphs show the distributions of allocated slots per user per drop for the EXP algorithm with 16 users per sector.

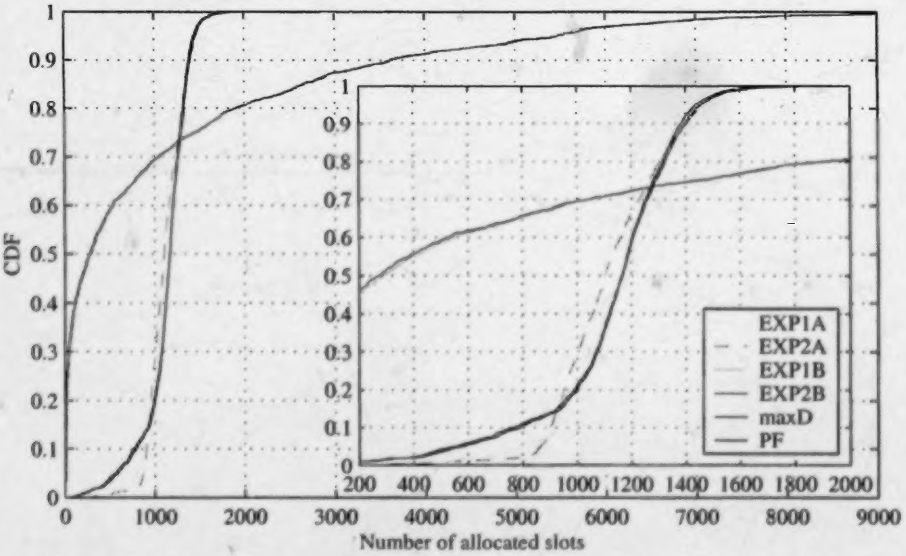


Figure 5.165: Distributions of allocated slots per user per drop for the EXP scheduling algorithm in the ITU pedestrian A channel with 16 users per sector

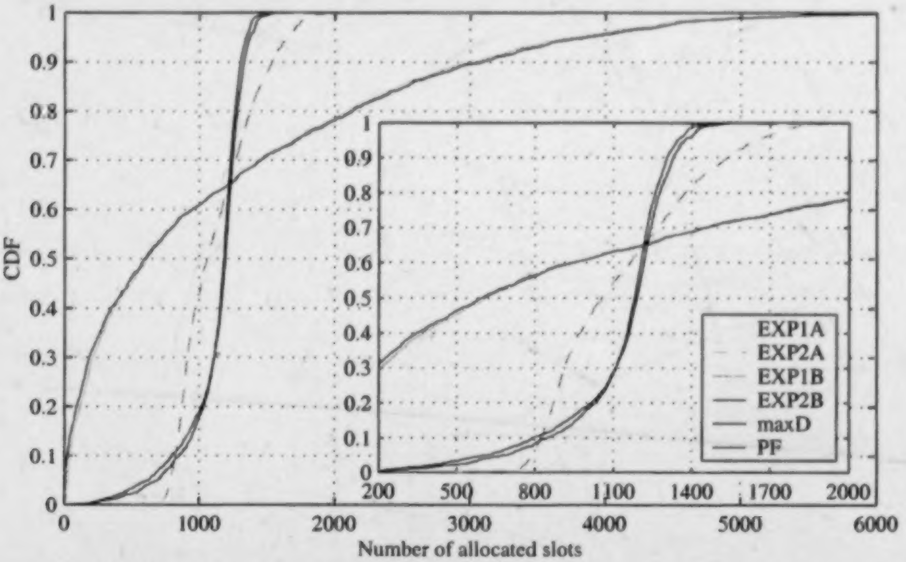


Figure 5.166: Distributions of allocated slots per user per drop for the EXP scheduling algorithm in the ITU pedestrian B channel with 16 users per sector

There is not a great deal of difference to be seen between the maxD and EXP1B distributions, nor between the PF and EXP2B algorithms. The main difference, which is not clear from the graphs, is that users are no longer starved for service under the EXP1B algorithm as they were under the maxD algorithm. They may only receive a few transmissions in some instances, but the number is still larger than zero. The EXP1A and EXP2A distributions are also fairly narrow. The following table compares the percentiles and standard deviations of the slot distributions.

TABLE XLIII:
PERCENTILES AND STANDARD DEVIATION OF SLOTS ALLOCATED PER USER PER DROP FOR THE EXPONENTIAL RULE SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN A AND B CHANNELS WITH 16 USERS PER SECTOR

Channel	ITU pedestrian A						ITU pedestrian B					
Algorithm / Percentile	EXP1A	EXP2A	EXP1B	EXP2B	maxD	PF	EXP1A	EXP2A	EXP1B	EXP2B	maxD	PF
10th	938	902	14	792	0	766	925	825	43	849	21	797
25th	1024	976	43	1044	22	1040	981	893	167	1077	140	1075
50th (median)	1096	1104	292	1167	275	1163	1080	1058	612	1182	597	1190
75th	1188	1266	1449	1281	1433	1290	1245	1323	1763	1242	1764	1258
90th	1325	1402	3589	1385	3628	1401	1401	1536	3043	1304	3085	1328
Std. Dev.	249.0	201.0	1759.5	256.1	1794.6	265.9	188.1	276.3	1251.5	206.8	1280.3	231.9

Overall, it was seen that using units of slots for the delay in the metric of the EXP algorithm produced very poor results. Delay was overemphasized to the near exclusion of channel conditions, and the resulting throughput, delay, and fairness was similar to the round robin algorithm. Using units of seconds for the delay produced much better results. The "B" set of simulations for the EXP algorithm produced results very similar to the non-EXP counterparts of the algorithms. There was a slight sacrifice in throughput and average delay per user and per packet. In return, the chances of a user or packet experiencing an extremely long delay dropped dramatically, especially for the EXP1B algorithm. The instances of a user being starved for service also dropped to zero. The distributions became slightly narrower as the users shared the resources a bit more fairly. Thus, the exponential rule proved to be a useful modification to the maxD and PF algorithms.

5.3.8. Maximum Rate / Proportionally Fair / Exponential Rule Hybrid (maxD/PF/EXP)

It was seen in the previous section that the multiplication of the exponential term involving the HOL delay to the maxD and PF metrics improved the delay characteristics of the algorithm without a significant sacrifice in the other statistics. Therefore, it would be a trivial problem to add that term to other algorithms to improve their delay response. Specifically, the exponential term could multiply the metric from the maxD/PF hybrid algorithm discussed earlier to create a tunable scheduling algorithm with an improved delay response. This maxD/PF/EXP hybrid was only analyzed using units of seconds for the delay, since it was seen that units of slots performed very poorly.

The following two figures show the average sector throughput per drop as a function of the number of users per sector for the maxD/PF/EXP hybrid scheduling algorithm.

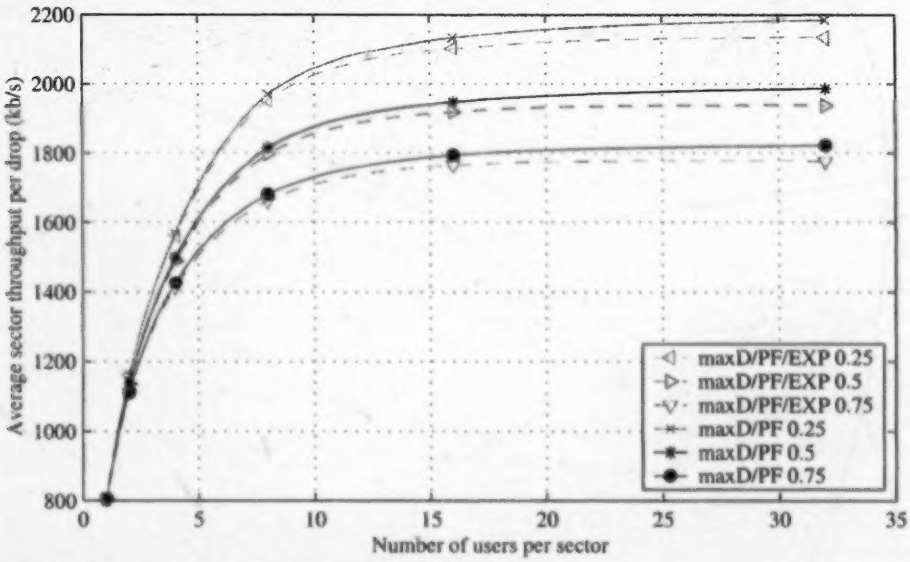


Figure 5.167: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian A channel for the maxD/PF/EXP scheduling algorithm

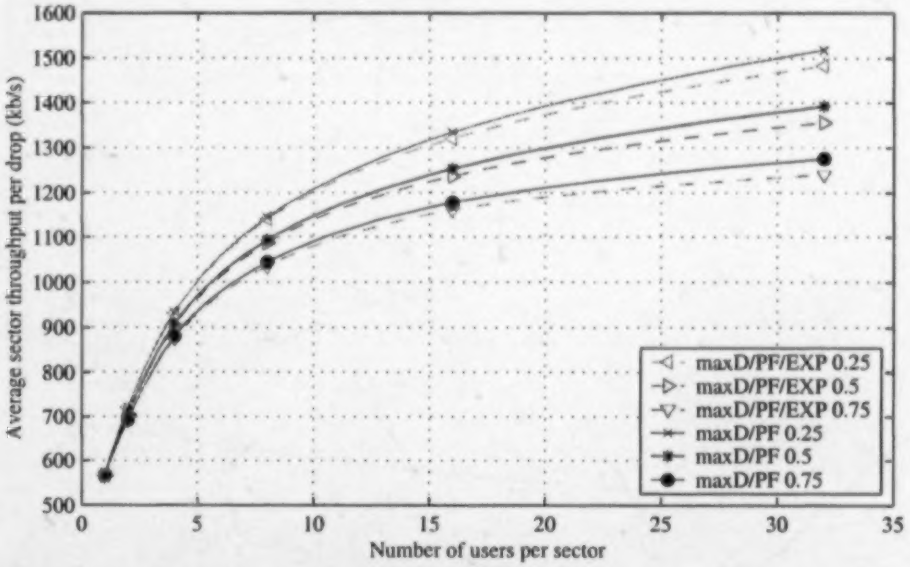


Figure 5.168: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian B channel for the maxD/PF/EXP scheduling algorithm

It can be seen that there still is not a significant reduction in throughput by adding the exponential delay term. In addition, even with the added term, the tuning parameter *a* still linearizes the throughput range between that of the EXP1B algorithm and that of the EXP2B algorithm.

The next two figures show the distributions of throughput per user per drop with 16 users per sector for the maxD/PF/EXP algorithm.

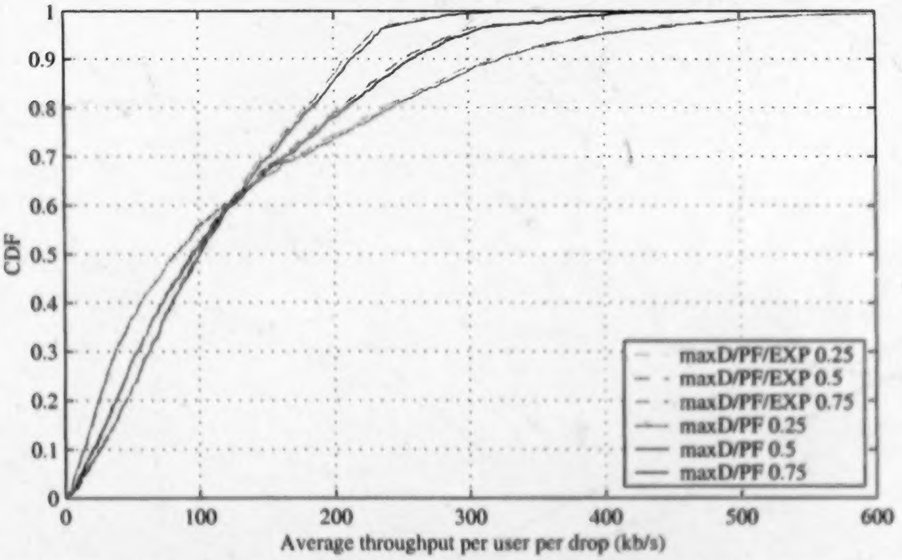


Figure 5.169: Distributions of average throughput per user per drop for the maxD/PF/EXP scheduling algorithm in the ITU pedestrian A channel with 16 users per sector

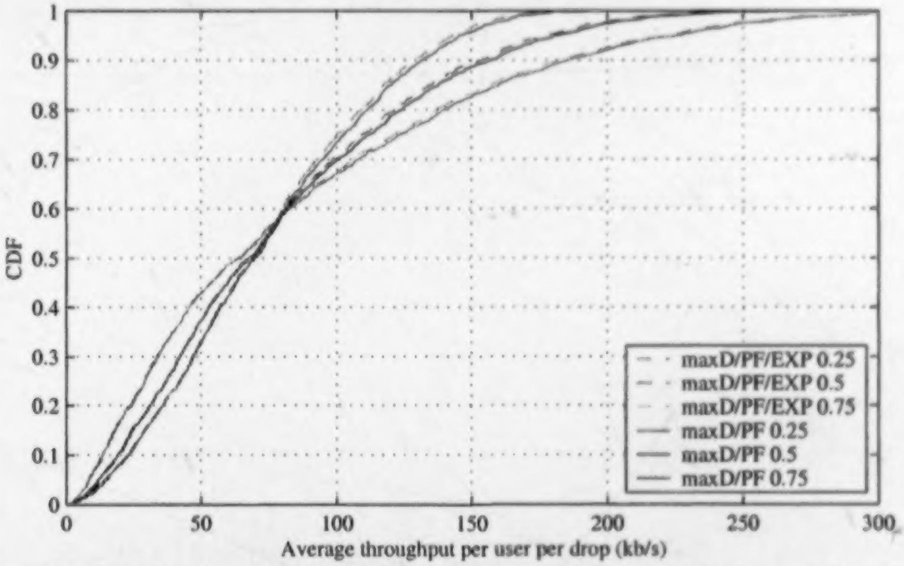


Figure 5.170: Distributions of average throughput per user per drop for the maxD/PF/EXP scheduling algorithm in the ITU pedestrian B channel with 16 users per sector

As with the EXP1B and EXP2B algorithms, apart from a minor tightening of the distributions, there is virtually no difference to be seen in the above graphs. The following table quantitatively confirms this similarity.

TABLE XLIV:
STATISTICS FOR AVERAGE THROUGHPUT PER USER PER DROP DISTRIBUTIONS FOR THE MAXD/PF/EXP SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN A AND B CHANNELS WITH 16 USERS PER SECTOR

Channel	ITU pedestrian A	ITU pedestrian B

Algorithm	maxD/ PF/EXP 0.25	maxD/ PF/EXP 0.5	maxD/ PF/EXP 0.75	maxD/ PF 0.25	maxD/ PF 0.5	maxD/ PF 0.75	maxD/ PF/EXP 0.25	maxD/ PF/EXP 0.5	maxD/ PF/EXP 0.75	maxD/ PF 0.25	maxD/ PF 0.5	maxD/ PF 0.75
Mean (kb/s)	131.5	119.9	110.3	133.3	121.7	112.1	82.5	77.3	72.6	83.4	78.3	73.6
Median (kb/s)	82.6	93.9	98.8	82.9	95.1	100.2	63.7	67.5	69.0	63.9	67.6	69.6
Std. Dev. (kb/s)	129.4	91.5	68.7	132.2	93.9	70.6	67.7	51.2	39.3	69.0	52.7	40.6

The following diagrams show the distributions of effective bit rates for packets sent under the maxD/PF/EXP algorithm with 16 users per sector.

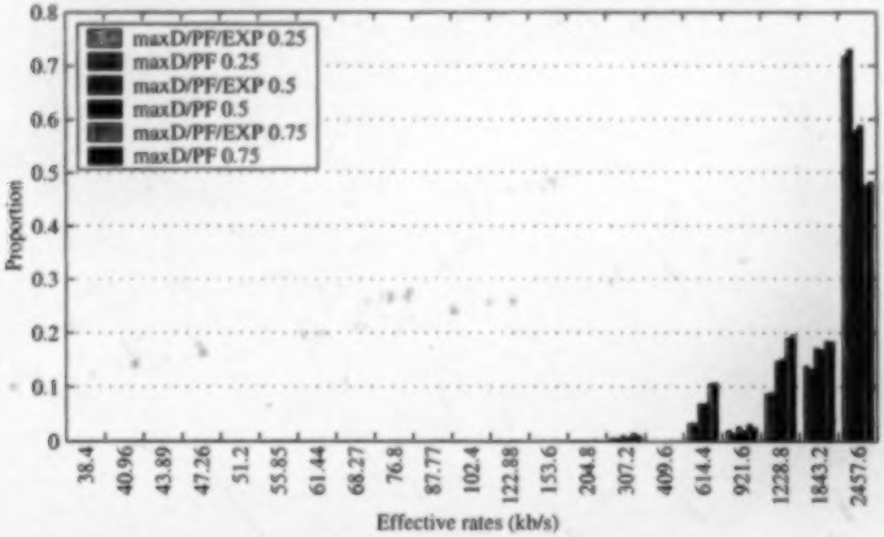


Figure 5.171: Distributions of effective bit rates for packets transmitted under the maxD/PF/EXP scheduling algorithm in the ITU pedestrian A channel with 16 users per sector

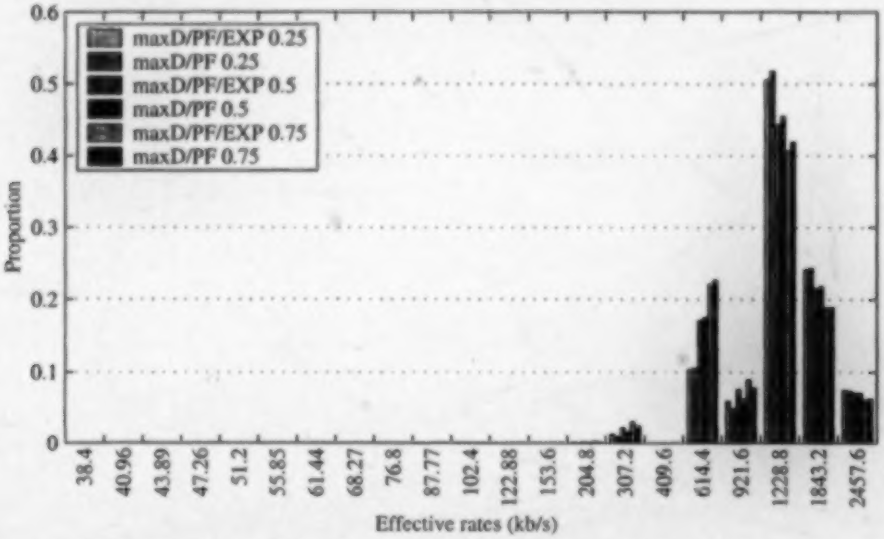


Figure 5.172: Distributions of effective bit rates for packets transmitted under the maxD/PF/EXP scheduling algorithm in the ITU pedestrian B channel with 16 users per sector

Once again, there is very little difference to be seen between the maxD/PF algorithm and the maxD/PF/EXP algorithm. The change in the proportions as the tuning parameter is increased still appears to be more or less linear.

The next two figures show the distributions of formats and transmissions used to send packets via the maxD/PF/EXP algorithm with 16 users per sector.

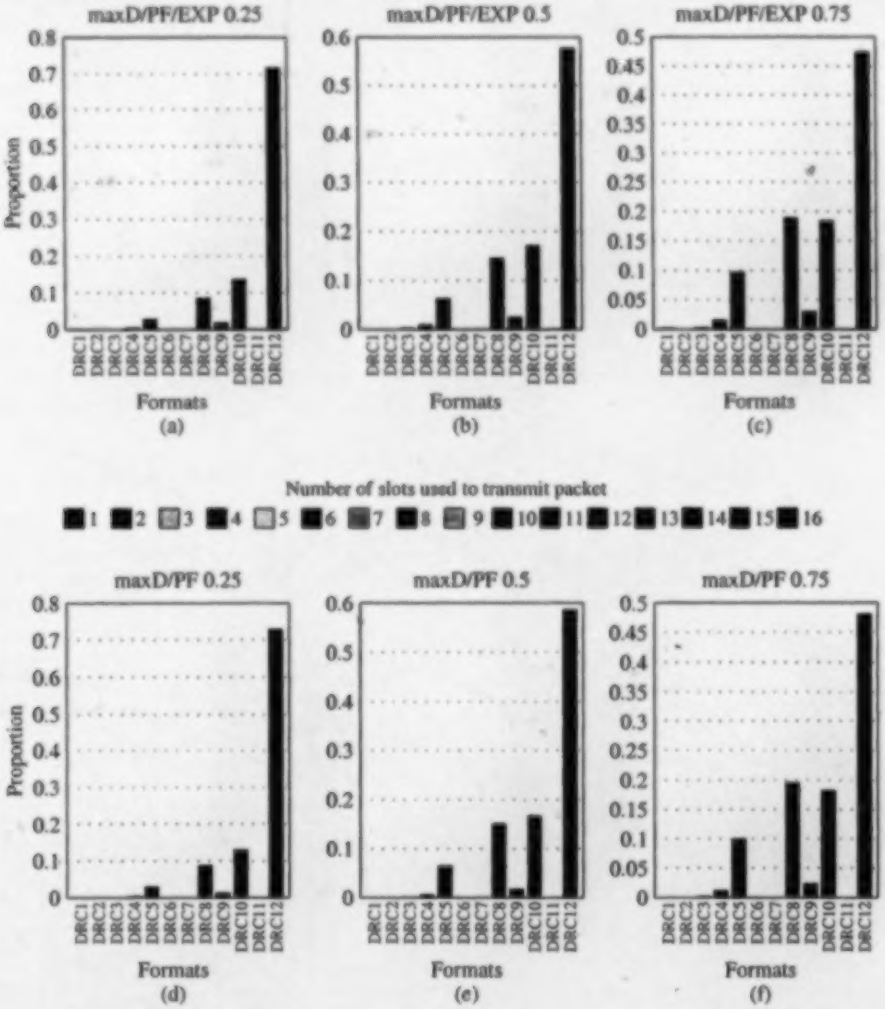


Figure 5.173: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a)-(c) maxD/PF/EXP, and (d)-(f) maxD/PF scheduling algorithms in the ITU pedestrian A channel with 16 users per sector

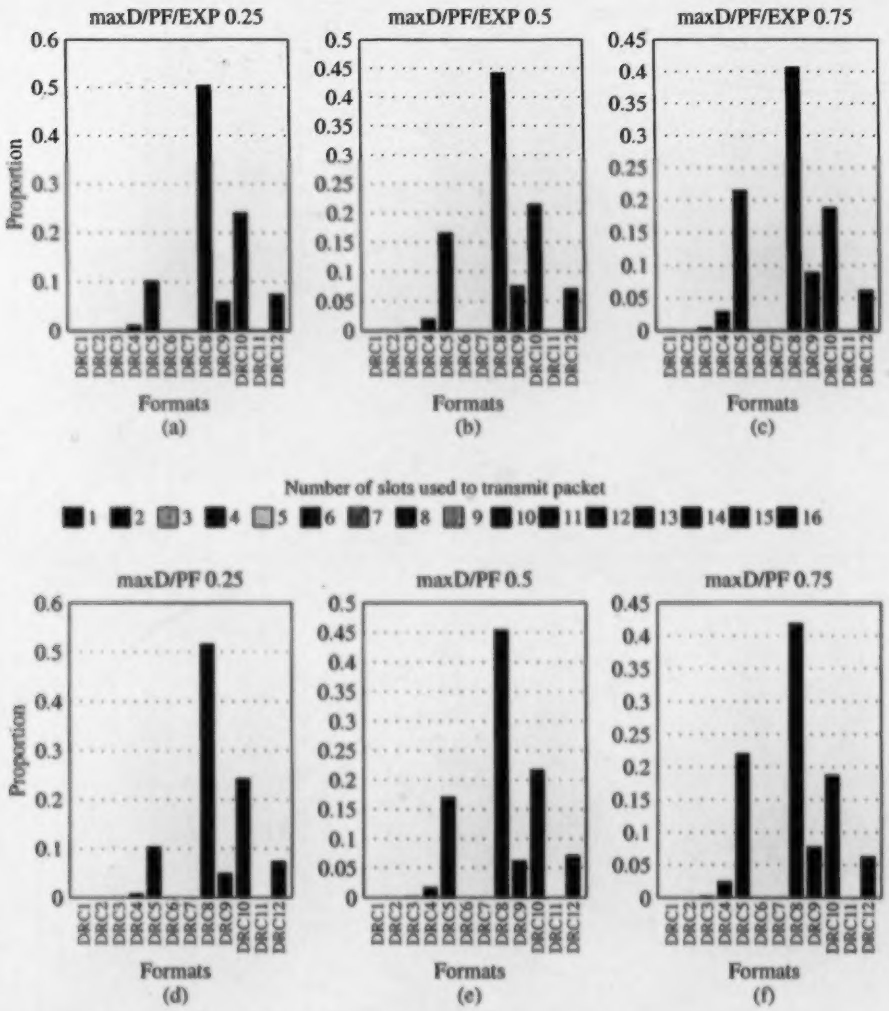


Figure 5.174: Distributions of formats used for correctly received packets and number of slots used to receive the packets for the (a)-(c) maxD/PF/EXP, and (d)-(f) maxD/PF scheduling algorithms in the ITU pedestrian B channel with 16 users per sector

There is a slight shift in proportions from the highest rate formats to lower rate formats between the maxD/PF algorithm and the maxD/PF/EXP algorithm. However, as is the case of the EXP algorithm, the change is very minor and the distributions are still very much the same.

The following two diagrams show the average delay per packet per drop as a function of the number of users per sector for the maxD/PF/EXP hybrid.

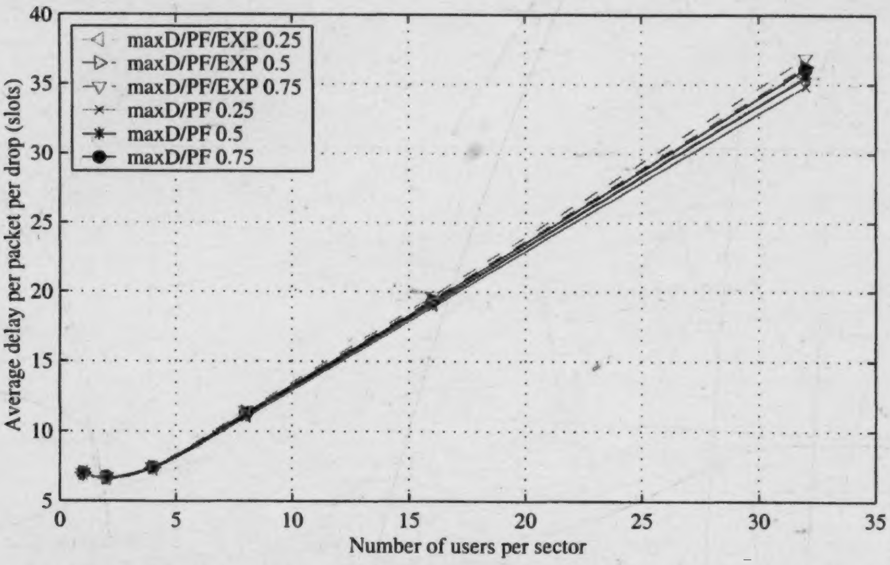


Figure 5.175: Average delay per packet per drop versus number of users per sector for the maxD/PF/EXP scheduling algorithm in the ITU pedestrian A channel

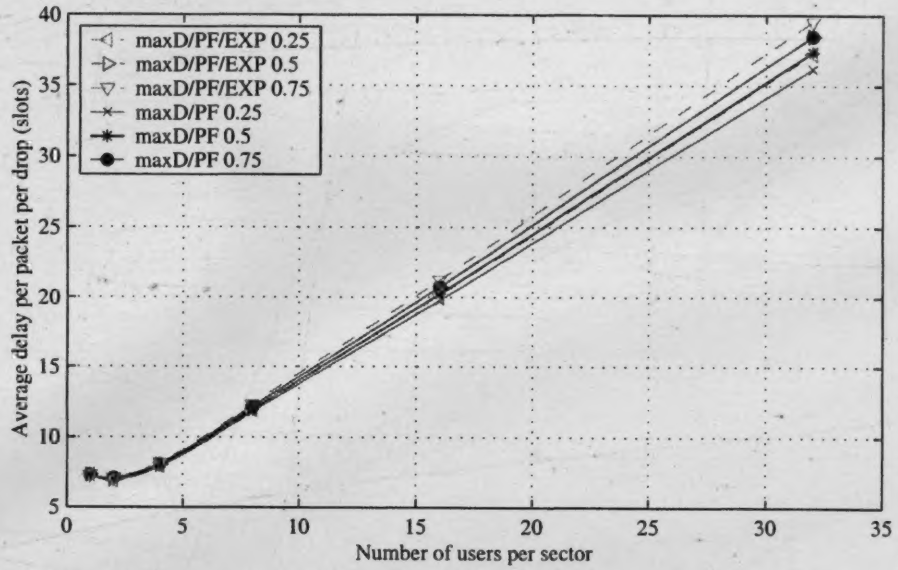


Figure 5.176: Average delay per packet per drop versus number of users per sector for the maxD/PF/EXP scheduling algorithm in the ITU pedestrian B channel

The above graphs are a bit difficult to read since the delay curves are all close together. There is an increase in the average delay per packet seen when moving from the maxD/PF algorithm to the maxD/PF/EXP hybrid algorithm. Overall, however, the change is nowhere near as great as that seen when moving from the maxD algorithm to the EXP1B algorithm. The increase is more along the lines of the added delay moving from the PF algorithm to the EXP2B algorithm. At 32 users per sector, the extra delay on average is only about one or two slots, which isn't really significant. Therefore, there still has yet to be seen any marked difference between the maxD/PF/EXP algorithm and the maxD/PF algorithm.

The next two figures show the average packet delay per user per drop as a function of the number of users per sector for the maxD/PF/EXP algorithm.

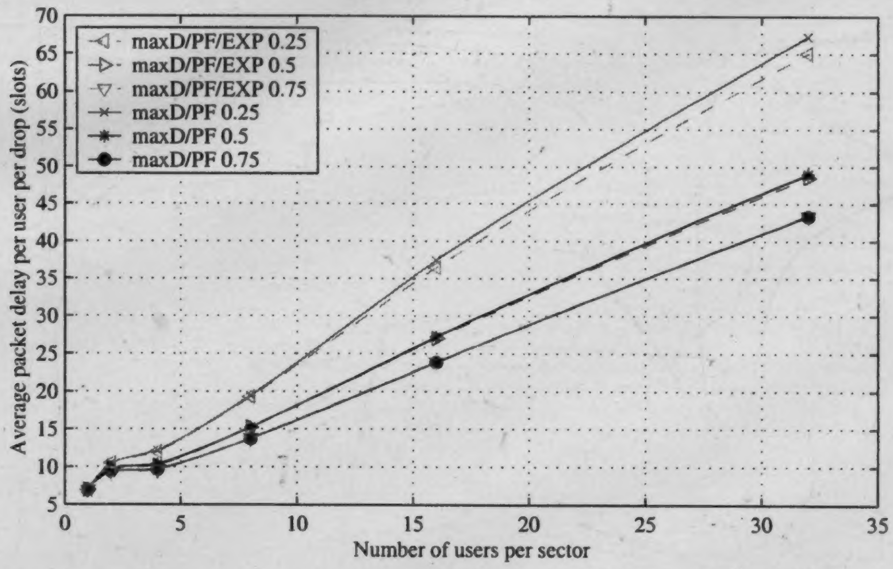


Figure 5.177: Distributions of average packet delay per user for the maxD/PF/EXP scheduling algorithm in ITU pedestrian A channel with 16 users per sector

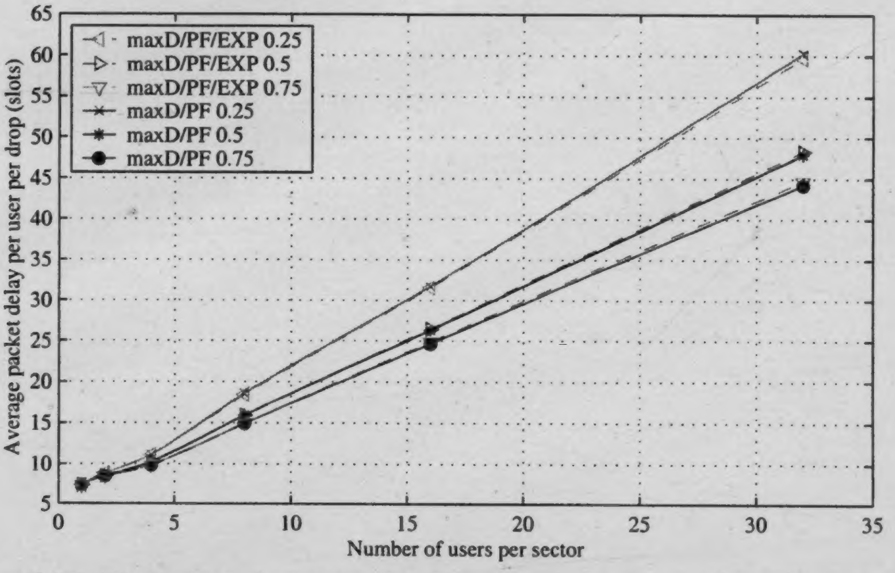


Figure 5.178: Distributions of average packet delay per user for the maxD/PF/EXP scheduling algorithm in ITU pedestrian B channel with 16 users per sector

There is a visible reduction in delay per user that can be seen for the maxD/PF/EXP algorithm in the ITU pedestrian A channel with a tuning parameter value of 0.25. Still, like everything seen so far with the maxD/PF/EXP algorithm, the change is rather inconsequential, with a maximum lessening on average of about 2 slots at 32 users per sector. There was virtually no change in the average delay per user for the other cases of the maxD/PF/EXP algorithm.

The following figures show the distributions of delays per packet for the maxD/PF/EXP algorithm with 16 users per sector.

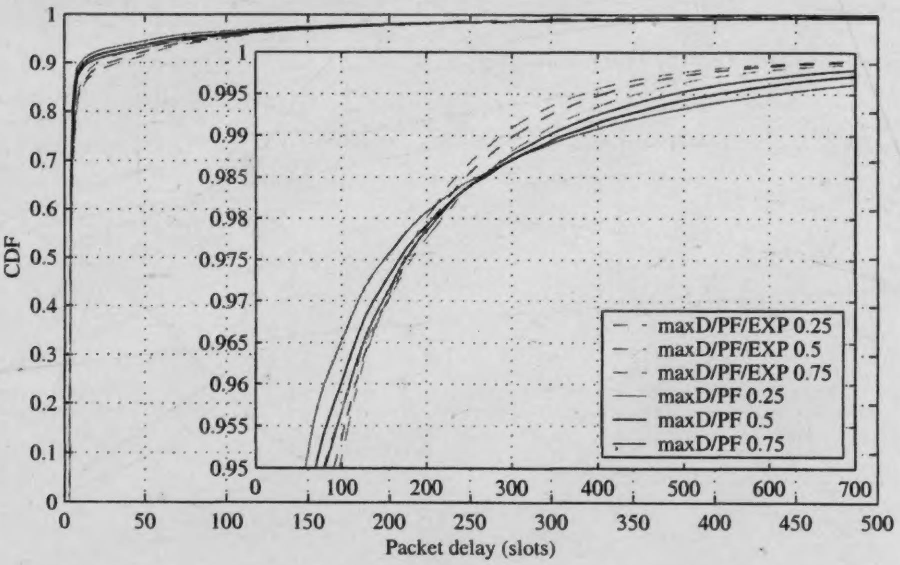


Figure 5.179: Cumulative distribution functions of delays experienced by packets under the maxD/PF/EXP scheduling algorithm in the ITU pedestrian A channel with 16 users per sector

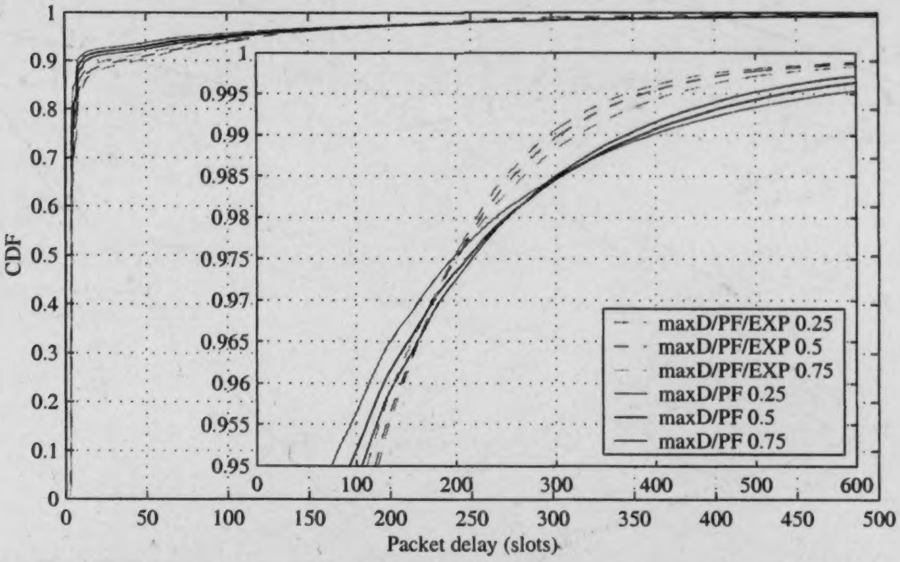


Figure 5.180: Cumulative distribution functions of delays experienced by packets under the maxD/PF/EXP scheduling algorithm in the ITU pedestrian B channel with 16 users per sector

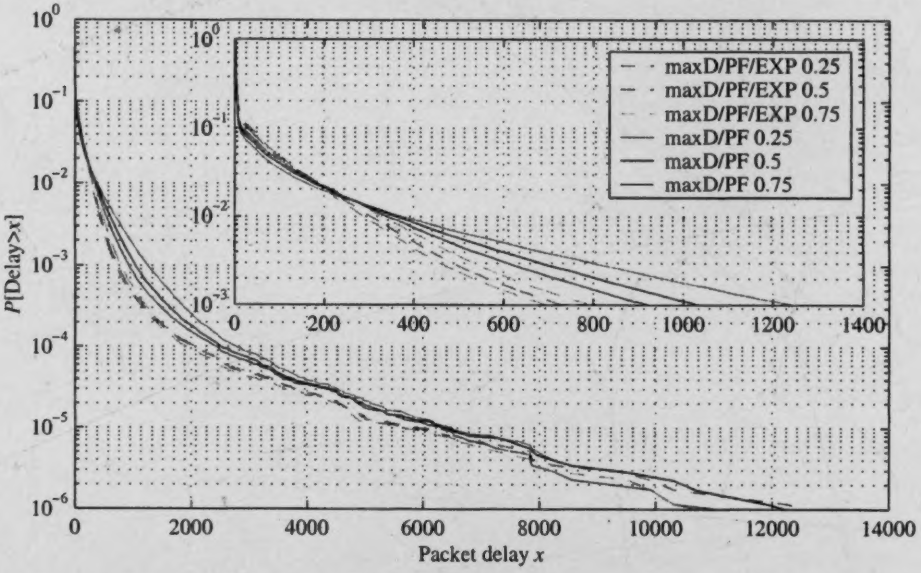


Figure 5.181: Probability of delays experienced by packets being larger than a given value under the maxD/PF/EXP scheduling algorithm in the ITU pedestrian A channel with 16 users per sector

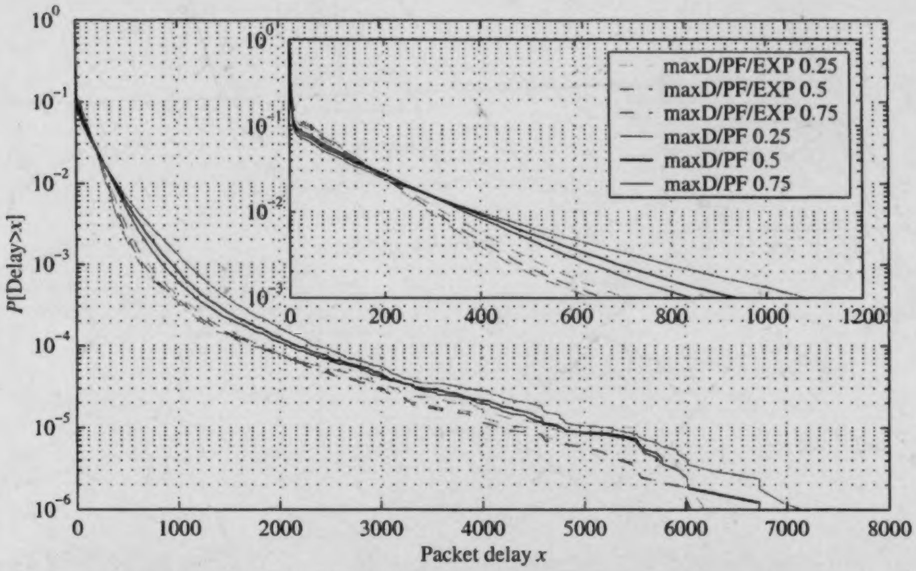


Figure 5.182: Probability of delays experienced by packets being larger than a given value under the maxD/PF/EXP scheduling algorithm in the ITU pedestrian B channel with 16 users per sector

The above distributions finally show a noteworthy change in the delays experienced by packets. The is a significant reduction in the probability of packets experiencing very long delays using the EXP extension in the maxD/PF/EXP algorithm than in using the maxD/PF algorithm alone. There is also a decrease in the probability of a packet experiencing an extremely short delay, as packets that would normally be transmitted rapidly in groups are delayed somewhat to compensate for the lengthy delays experienced by other packets. Fortunately, that extra delay the packets of users in good channel conditions experience is much smaller than the reduction in the wait that the delayed packets of users in poorer channel conditions gain.

The following table compares the percentiles and standard deviations of the above packet delay distributions.

TABLE XLV:
PERCENTILES AND STANDARD DEVIATIONS OF PACKET DELAYS FOR THE MAXD/PF/EXP SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN A AND B CHANNELS WITH 16 USERS PER SECTOR

Channel	ITU pedestrian A						ITU pedestrian B					
Algorithm / Percentile	maxD/ PF/EXP 0.25	maxD/ PF/EXP 0.5	maxD/ PF/EXP 0.75	maxD/ PF 0.25	maxD/ PF 0.5	maxD/ PF 0.75	maxD/ PF/EXP 0.25	maxD/ PF/EXP 0.5	maxD/ PF/EXP 0.75	maxD/ PF 0.25	maxD/ PF 0.5	maxD/ PF 0.75
50th (median)	4	4	4	4	4	4	4	4	4	4	4	4
90th	21.5	28.3	34.9	10.1	12.4	15.5	25.7	36.2	45.1	8.0	11.0	13.6
95th	78.9	90.7	97.2	57.9	69.5	80.5	110.3	118.5	121.5	74.8	92.8	105.2
99th	328.2	302.6	287.2	372.7	359.6	340.9	322.7	303.9	293.2	395.9	385.5	370.4
99.5th	444.0	405.8	382.6	606.6	537.3	485.0	417.4	388.2	371.0	576.7	527.8	491.9
99.9th	793.3	732.0	697.3	1245.8	1037.0	922.2	707.0	656.5	633.0	1085.6	937.6	843.0
99.95th	1018.1	975.5	930.0	1607.8	1338.0	1190.6	917.4	874.3	850.6	1362.2	1162.2	1069.3
99.99th	2068.3	2034.6	1927.4	2783.6	2516.8	2442.6	1820.7	1760.7	1722.3	2348.8	2170.0	2087.2
Std. Dev.	77.23	74.50	71.13	98.76	90.11	84.42	71.06	67.98	67.23	90.09	83.5	79.93

Both the upper percentile delays and the standard deviations are smaller for the maxD/PF/EXP algorithm, indicating that there is more fairness in how packets are dealt with over the maxD/PF algorithm.

The following two diagrams show the distributions of average packet delay per user per drop for the maxD/PF/EXP algorithm with 16 users per sector.

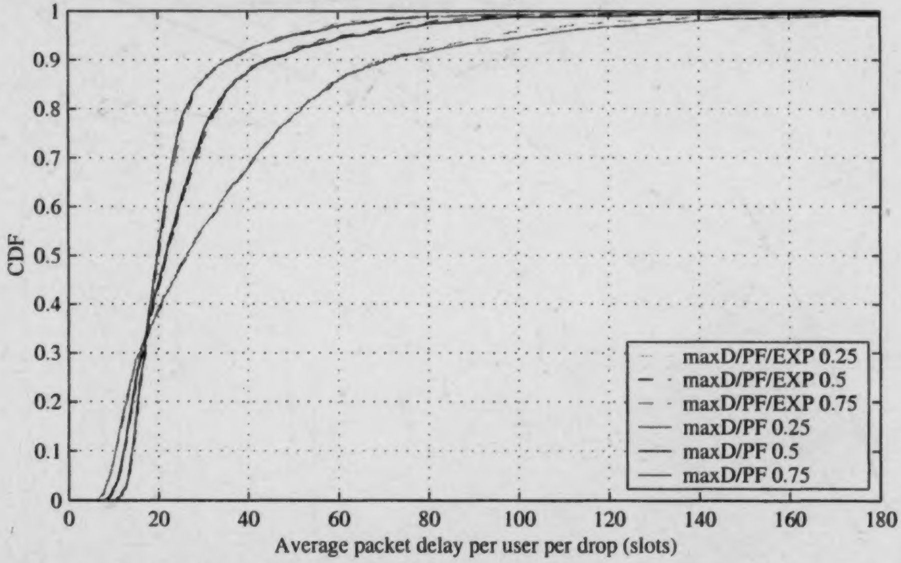


Figure 5.183: Distributions of average packet delay per user for the maxD/PF/EXP scheduling algorithm in ITU pedestrian A channel with 16 users per sector

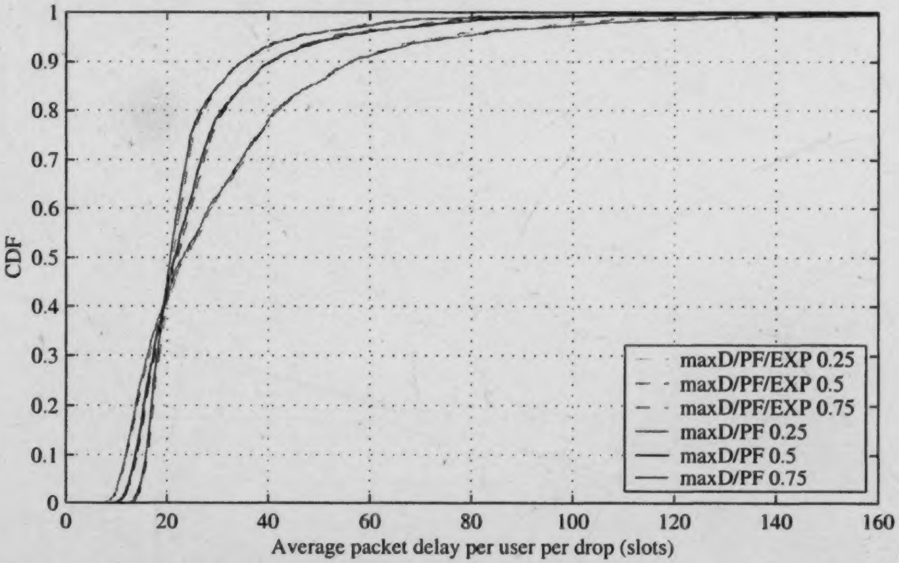


Figure 5.184: Distributions of average packet delay per user for the maxD/PF/EXP scheduling algorithm in ITU pedestrian B channel with 16 users per sector

Once again, there is no discernable difference to be seen in the distributions of delay per user; the curves of the maxD/PF/EXP algorithm and the maxD/PF algorithm lie almost on top of each other. Consequently, there is not much point in comparing the percentiles and standard deviations of the distributions, as it would not yield any new information.

The following two figures show the distributions of allocated slots per user per drop for the maxD/PF/EXP algorithm with 16 users per sector.

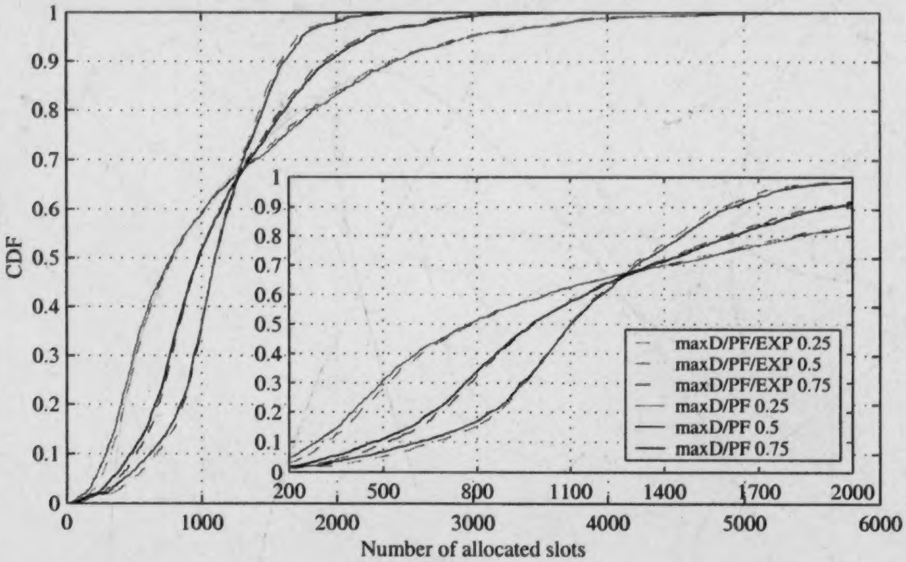


Figure 5.185: Distributions of allocated slots per user per drop for the maxD/PF/EXP scheduling algorithm in the ITU pedestrian A channel with 16 users per sector

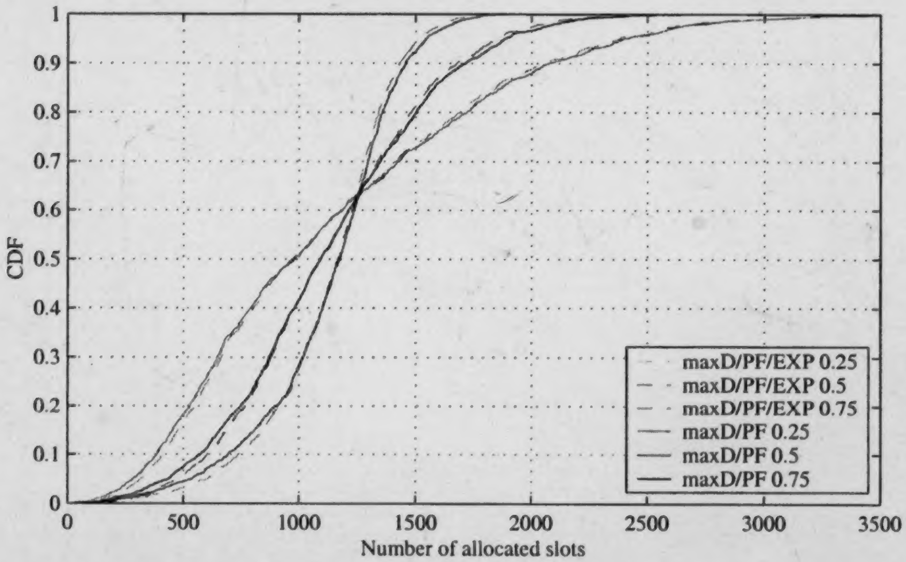


Figure 5.186: Distributions of allocated slots per user per drop for the maxD/PF/EXP scheduling algorithm in the ITU pedestrian B channel with 16 users per sector

There is a minor tightening of the slot allocation distributions for the maxD/PF/EXP algorithm. The following table gives a quantitative feel of this narrowing.

TABLE XLVI:
PERCENTILES AND STANDARD DEVIATION OF SLOTS ALLOCATED PER USER PER DROP FOR THE MAXD/PF/EXP SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN A AND B CHANNELS WITH 16 USERS PER SECTOR

Channel	ITU pedestrian A						ITU pedestrian B					
Algorithm / Percentile	maxD/ PF/EXP 0.25	maxD/ PF/EXP 0.5	maxD/ PF/EXP 0.75	maxD/ PF 0.25	maxD/ PF 0.5	maxD/ PF 0.75	maxD/ PF/EXP 0.25	maxD/ PF/EXP 0.5	maxD/ PF/EXP 0.75	maxD/ PF 0.25	maxD/ PF 0.5	maxD/ PF 0.75
10th	324	514	667	289	473	617	414	606	742	386	578	698
25th	473	733	914	451	718	907	609	826	979	590	818	971
50th (median)	785	983	1100	775	976	1101	983	1093	1166	973	1093	1175
75th	1604	1469	1371	1623	1487	1387	1552	1415	1313	1574	1425	1326
90th	2393	1903	1584	2419	1945	1615	2046	1677	1432	2077	1715	1457
Std. Dev.	897.7	565.8	367.0	926.0	592.5	392.8	657.4	420.6	279.1	678.2	444.1	303.5

As the graphs and the table show, there is no major difference to be seen on the slot distributions. The percentiles do not change by any more than 50 slots at any point. Thus, the two algorithms are essentially the same in terms of the number of slots each one allocates to the users in the sector.

Overall, there was no significant difference to be seen between most of the results of the maxD/PF/EXP algorithm and the maxD/PF algorithm. The exception was in the distribution of packet delays, where there was a noticeable reduction in the probability of a lengthy packet delay for the maxD/PF/EXP algorithm. Thus, with all else being equal, the maxD/PF/EXP algorithm is a definite improvement over the maxD/PF algorithm.

5.3.9. Conclusions

Overall, there were definite trends seen for all the analyzed scheduling algorithms. In general, if an algorithm attempted to increase the average sector throughput, there was a corresponding drop in the average delay per packet, but an increase in the average delay per user. The distributions of throughputs per user, delays per packet and per user, and slots also became more spread out. Conversely, if an algorithm tried to reduce the delay or improve the fairness of the system, there usually was a throughput reduction in tradeoff. The average delay per packet would increase, while the average delay per user would decrease. The distributions would become tighter as resources were shared more equally. However, decreasing the delay was only helpful up to a certain point. If the delay requirements became more important than the channel conditions, almost all of the results would suffer. The throughput would decrease, and both types of average delay would increase noticeably, since transmissions would occur at suboptimal times. The distributions would become more spread apart again in some cases. The only benefit was that the probability of a packet having a very long delay became rather small.

The maximum rate algorithm clearly was the best algorithm in terms of throughput, but it was also extremely unfair in terms of the amount of service users in poorer channel conditions received. The EXP1B algorithm improved somewhat on this fairness issue without much of a penalty in throughput. The maxD/PF and maxD/PF/EXP algorithms with low tuning parameter values also provided a decent amount of throughput. The minimum performance algorithm also yielded good throughput for low numbers of users per sector, but as the sector became more crowded, the throughput dropped off. In terms of delay, the M-LWDF 1 & 2, EXP1A, and EXP2A algorithms all showed very low probabilities of extremely long packet delays, but all four algorithms suffered from a major throughput reduction and a penalty in the average delays experienced by packets and by users. The proportionally fair, slot-wise fair, and EXP2B algorithms, and the maxD/PF and maxD/PF/EXP algorithms with high values for the tuning parameter all seemed to achieve a good balance between throughput, delay, and fairness. The packet-wise fair algorithm also yielded a decent performance, especially when compared to the round robin algorithm, which also attempted to provide users with equal numbers of packets. Graphs comparing the results of all the analyzed scheduling algorithms simultaneously for the ITU pedestrian A and B channels can be found in Appendix B.

One final note of interest can be seen in the distributions of formats and slots used for packets of each format. When packets were sent using the lowest rate formats, early terminations tended to occur quite often and usually after only a few re-transmissions. This can be attributed to the margins used to determine the supportable formats for each user. In order to obtain a 1% packet error for those formats, margins of several dB were required for the SNR. This meant that when transmissions did occur for those formats, they were very likely to end early. In comparison, the margins for the highest rate formats were only a fraction of a decibel.

Using perfect multislot prediction, the simulation can determine if an early termination would occur under a given format, and if so, the user could then possibly request a higher rate format. This would allow that user a greater chance of being selected to receive a packet. The effect of perfect multislot prediction on the throughput, delay, and fairness shall be examined in the next section.

5.4. Pedestrian Channels, Perfect Multislot Prediction

5.4.1. Effects of Perfect Multislot Prediction

In perfect multislot prediction, the exact level of the future SIR values (including imperfections as discussed in Chapter 4) is known at all times. Multislot prediction involves averaging those values over the slots where a transmission could potentially take place. For example, when checking if a rate of 38.4 kb/s can be supported, the 16 SIR values corresponding to the 16 slots where the transmission might occur are averaged. When testing if 921.6 kb/s can be supported, only 2 SIR values are averaged. That average value is then compared with the required E_S/N_0 in the AWGN channel to obtain a 1% packet error rate using the given format and bit rate. If the average is greater than that AWGN threshold, then the format is deemed to be supported. The highest rate for which the average SIR is greater than the threshold is the rate that each mobile requests from the base station. Note that this method effectively overestimates the SIR required for a format. Since the average SIR is greater than that required for a 1% PER, the actual PER will end up being less than 1%. Due to the steepness of the turbo code curves, this will often mean the PER is in fact an order or two lower than the desired 1% level. In reality, it would be desirable to subtract a small margin from the average SIR, to in effect overestimate the rate that can be supported. This would bring the PER up to the desired 1% level. Unfortunately, also due to the steep nature of the turbo code PER curves, the packet error rate showed an extreme sensitivity to small changes in the margin. As such, it was extremely difficult, if not impossible, to get the PER to stabilize around 1%. Therefore, it was decided to not use a margin and leave the packet error rates at their low values. This would not result in a large change in throughput, as discussed near the beginning of Chapter 5. It does however eliminate the need to iterate to find appropriate margins, making the simulations much shorter.

Figure C.1 and Figure C.2 in Appendix C show the average sector throughput as a function of the number of users per sector for the various scheduling algorithms in the ITU pedestrian A and B channels, respectively. The most noticeable observation from the graphs is that the throughput does not always increase; for some algorithms, the average sector throughput actually decreases. Whether the throughput goes up or down seems to depend on how strongly the metric for the scheduling algorithm takes into account channel conditions. At one extreme, there is a noticeable increase in the throughput for the maxD algorithm, particularly in the ITU pedestrian B channel. At the other extreme, the RR algorithm shows a decrease in throughput, largely in the ITU pedestrian A channel. Overall, the maxD, maxD/PF, PF, EXP1B, EXP2B, maxD/PF/EXP, and slot-fair algorithms all showed either an increase or no significant change in the throughput. The RR, M-LWDF 2, EXP1A, EXP2A, and packet-fair algorithms all displayed a decrease in sector throughput. The throughput of the M-LWDF 1 algorithm decreased in the ITU pedestrian A channel, but increased in the ITU pedestrian B channel. The minperf algorithm displayed an increase in throughput for low numbers of users per sector, but a decreased throughput for larger numbers of users per sector. The relationship between channel conditions and throughput change is correlated to the selection of formats used for each algorithm. In general, the more an algorithm emphasizes channel conditions (i.e. the requested rate), the more likely it is that users with high data rates will be selected. In other words, users who request low data rates are selected rather infrequently. In comparison, if additional elements are considered beyond the requested rate, users who request low data rates can be selected much more often. Consider now the margins used with outdated channel information. For the lowest data rates, those margins usually were several dB in magnitude in order to obtain the desired packet error rate of 1%. This results in two effects. First, the level of SIR required to not appear to be in outage is much higher than in an AWGN channel. For example, instead of having an SIR of at least –13.5 dB to request a non-zero rate, a user must instead have an SIR of around –8.5 to –4.5 dB. Second, when a user is selected to receive data at one of the low rates, because of the large margins, it was seen that the transmission was often received correctly after only a few slots. Hence, for the lower rate data users, they end up waiting a longer time to receive data, but when they do, their effective rates tend to be higher than the nominal rates. Compare this situation with perfect multislot prediction. Knowing the channel perfectly means knowing exactly if the user can support a non-zero data rate (to a maximum of 1% PER). Thus, users will be seen to be able to receive data more often. Second, knowing the channel perfectly means knowing exactly the maximum data rate that can be supported, and as such requesting that higher rate. Therefore, early terminations are less likely to occur. These two effects mean that users with who request low data rates will receive a lower throughput when receiving data using perfect multislot prediction than by using margins. By extension, algorithms that select low data rate users fairly often will see their throughputs drop under perfect multislot prediction. This explains the lower throughput for the RR, M-LWDF 2, EXP1A, EXP2A, and packet-fair algorithms. For the minperf algorithm, at low numbers of users per sector, the algorithm is able to provide the minimum specified throughput to all the users most of the time just by transmitting to the user with the highest requested rate. Thus, low rate formats are not used very often, and the throughput increases over the case with margins. For large numbers of users per sector, the minimum is less likely to be met for all users considering just the maximum requested rate, meaning more transmissions to weaker users at low data rates to meet that minimum, and consequently the throughput drops relative to the case when using margins. This same effect also shifts the location of where the throughput begins to decrease to a smaller number of users per sector.

One other observation is that the benefit of perfect multislot prediction seems to be higher in the ITU pedestrian B channel than in the A channel. Recall that the ITU pedestrian A channel changes more slowly over time. There would be a high correlation between the SIR values over the period of several slots. As such, perfect knowledge of the future channel conditions does not provide much more insight than knowledge of the current channel conditions. This is reflected in that the algorithms that do show a gain in throughput do not show much of a gain, and that the throughput of certain other algorithms does not significantly change at all. For the algorithms that drop in throughput, the drop is comparatively larger. In contrast, the ITU pedestrian B channel contains multipath components, and as such varies more rapidly. In this case, there is a larger benefit to perfect knowledge of the channel, as seen by the substantial increase in the throughput of the maxD and other algorithms. In addition, for those algorithms that show a decrease in throughput, the decrease is not as severe as in the ITU pedestrian A channel. Consider the case of the M-LWDF 1 algorithm. In the ITU pedestrian A channel, the throughput drops between using margins and using perfect multislot prediction. However, in the ITU pedestrian B channel, the throughput rises instead. The added benefit of the perfect prediction in the B channel counteracts the loss due to low rate users just enough to see an overall gain in throughput.

A last point of interest is that the throughput with 1 user per sector increases by an ample amount. In effect, any algorithm with one user per sector reduces to the maximum rate algorithm. The algorithm always selects the user with the largest requested rate. It just so happens that the user with the largest rate is always the same user, because there is only the one user to choose from. Therefore, having a better knowledge of what rate that user can support and using that maximum rate will simply increase the capacity of the channel/user. Interestingly, the proportion by which the throughput increases for 1 user per sector by using perfect multislot prediction is always larger than with 32 users per sector. Therefore, while the absolute sector throughput may increase or decrease by using perfect prediction, the throughput gain from 1 to 32 users in fact always decreases, regardless of the channel or scheduling algorithm.

Figures C.3 and C.4 in Appendix C show the distributions of average throughput per user per drop with 16 users per sector for the various scheduling algorithms in the ITU pedestrian A and B channels, respectively. In the ITU pedestrian A channel, there is a general shift downwards in the throughputs achieved by users and a corresponding tightening of the distributions. This results once again from how the perfect multislot prediction allows users with lower data rates to be selected more often. In the ITU pedestrian B channel, some of the distributions shift upwards in throughput, while others shift downward, with the trend being to match the direction of the shift in average sector throughput. The following tables compare the statistics of the distributions between using margins and using perfect prediction.

TABLE XLVII:
MEAN, MEDIAN, AND STANDARD DEVIATION OF AVERAGE THROUGHPUT PER USER PER DROP DISTRIBUTIONS FOR SCHEDULING ALGORITHMS IN THE ITU PEDESTRIAN A CHANNEL WITH 16 USERS/SECTOR COMPARING USE OF MARGINS WITH PERFECT MULTISLOT PREDICTION

Value	Using Margins / Using Perfect Prediction					
Algorithm	RR	maxD	PF	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75
Mean (kb/s)	40.7 / 32.1	145.7 / 148.6	103.6 / 103.0	133.3 / 134.2	121.7 / 121.8	112.1 / 111.4
Median (kb/s)	33.7 / 29.3	33.4 / 38.3	99.6 / 101.3	82.9 / 90.3	95.1 / 100.7	100.2 / 102.3
Std. Dev. (kb/s)	25.6 / 15.6	236.8 / 232.2	55.6 / 51.1	132.2 / 123.3	93.9 / 86.7	70.6 / 64.6
Algorithm	M-LWDF 1	M-LWDF 2	slot-fair	packet-fair	minperf 5 kb/s	minperf 10 kb/s
Mean (kb/s)	89.3/ 85.7	69.8 / 59.8	103.4 / 105.6	89.2 / 77.5	132.1 / 128.6	119.8 / 108.1
Median (kb/s)	55.4 / 60.0	60.5 / 54.6	106.9 / 111.4	90.1 / 78.8	32.7 / 35.0	33.0 / 33.7
Std. Dev. (kb/s)	88.9 / 75.5	43.5 / 32.9	52.0 / 50.6	40.1 / 31.2	205.6 / 191.9	178.1 / 153.5
Algorithm	minperf 15 kb/s	minperf 20 kb/s	minperf 25 kb/s	EXP1A	EXP2A	EXP1B
Mean (kb/s)	107.0 / 89.7	93.4 / 73.9	81.2 / 63.3	58.9 / 42.2	50.5 / 37.4	145.1 / 147.7
Median (kb/s)	33.7 / 32.5	35.1 / 33.1	38.1 / 36.2	39.3 / 34.2	42.6 / 33.5	35.2 / 40.9
Std. Dev. (kb/s)	149.5 / 119.5	119.4 / 88.2	91.6 / 63.9	55.3 / 29.4	31.0 / 19.7	232.9 / 227.2
Algorithm	EXP2B	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75		
Mean (kb/s)	102.5 / 101.5	131.5 / 132.4	119.9 / 120.0	110.3 / 109.8		
Median (kb/s)	98.5 / 99.8	82.6 / 89.1	93.9 / 99.4	98.8 / /101.1		
Std. Dev. (kb/s)	54.7 / 50.2	129.4 / 121.2	91.5 / 85.0	68.7 / 63.4		

TABLE XLVIII:
MEAN, MEDIAN, AND STANDARD DEVIATION FOR SCHEDULING ALGORITHMS IN THE ITU PEDESTRIAN B CHANNEL WITH 16 USERS/SECTOR COMPARING USE OF MARGINS WITH PERFECT MULTISLOT PREDICTION

Value	Using Margins / Using Perfect Prediction					
Algorithm	RR	maxD	PF	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75
Mean (kb/s)	26.7 / 26.2	87.7 / 96.6	69.5 / 71.6	83.4 / 90.1	78.3 / 83.4	73.6 / 77.2
Median (kb/s)	26.5 / 24.7	41.0 / 43.2	69.6 / 71.5	63.9 / 67.0	67.6 / 71.9	69.6 / 72.9
Std. Dev. (kb/s)	8.95 / 9.67	106.5 / 119.3	32.5 / 32.1	69.0 / 75.1	52.7 / 55.4	40.6 / 41.3
Algorithm	M-LWDF 1	M-LWDF 2	slot-fair	packet-fair	minperf 5 kb/s	minperf 10 kb/s
Mean (kb/s)	48.8 / 50.6	41.9 / 39.6	69.9 / 75.1	59.6 / 57.8	80.4 / 85.9	72.3 / 73.3
Median (kb/s)	41.4 / 41.8	41.2 / 38.7	67.7 / 71.8	58.2 / 56.3	38.2 / 38.7	35.9 / 34.5
Std. Dev. (kb/s)	33.6 / 35.3	18.4 / 17.3	35.1 / 38.6	24.4 / 22.8	92.5 / 101.2	77.3 / 81.0
Algorithm	minperf 15 kb/s	minperf 20 kb/s	minperf 25 kb/s	EXP1A	EXP2A	EXP1B
Mean (kb/s)	62.7 / 60.9	53.8 / 51.4	47.1 / 45.5	32.2 / 29.8	30.1 / 28.2	87.5 / 96.0
Median (kb/s)	33.5 / 31.4	32.2 / 30.5	34.5 / 33.5	30.1 / 27.4	29.4 / 26.5	41.9 / 45.0
Std. Dev. (kb/s)	60.3 / 61.4	44.3 / 44.4	30.2 / 30.1	15.3 / 13.9	11.1 / 11.2	105.0 / 116.5
Algorithm	EXP2B	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75		
Mean (kb/s)	68.4 / 70.5	82.5 / 88.8	77.3 / 82.2	72.6 / 76.0		
Median (kb/s)	68.5 / 70.6	63.7 / 66.6	67.5 / 71.0	69.0 / 71.9		
Std. Dev. (kb/s)	31.5 / 31.5	67.7 / 73.9	51.2 / 54.2	39.3 / 40.5		

In the ITU pedestrian A channel with perfect prediction, the standard deviation decreases and the distributions become tighter as users who request lower data rates are served more often, which results in less variation in the throughput received by each user. For the ITU pedestrian B channel, the standard variation instead increases in most cases, because although the lower rate users are served more often, the better prediction also increases the throughputs of intermediate to high rate users, resulting in more disparity in the throughputs received by each user. In general, the change in the mean and median throughputs per user reflect the change in the average sector throughputs; if the sector throughput increases, decreases, or stays about the same for a scheduling algorithm in a given channel, the mean and median throughputs per user do the same thing.

For the minperf algorithm, the probability of not meeting the minimum specified throughput is 0.81%, 1.44%, 1.75%, 2.81%, and 3.63% in the ITU pedestrian A channel and 0.25%, 0.44%, 1.06%, 1.63%, and 3.19% in the ITU pedestrian B channel for the five tested minimums respectively in increasing order. These probabilities are lower than the case with margins, so the fact that the lower rate users receive more service does provide an improvement in that respect.

Similarly, the probability of a user not receiving any service under the maxD algorithm is 11.56% and 3.69% for the ITU pedestrian A and B channels, respectively. This probability is lower for the A channel, but higher for the B channel. In the A channel, recall that transmissions at the highest rate were already quite prevalent with the use of margins. This was due in part to the smaller variations of the channel and the subsequent smaller margins required to obtain a 1% packet error rate. Removing those margins and using perfect prediction allowed a few more users in the sector to request the highest rate and more often. This subsequently reduced the probability of no service. However, in the ITU pedestrian B channel, the highest rate was originally requested much less often compared to the ITU pedestrian A channel. The larger variations in signal strength required larger margins for a 1% packet error rate. Removal of those margins and predicting the channel perfectly instead wound up causing the few users who requested the highest rate to request it more often. This consequently shifted service away from other users, in some cases completely. Hence, the probability of no service in fact increased in the ITU pedestrian B channel when using perfect prediction.

Figures C.5 and C.6 in Appendix C show the distributions of effective bit rates for packets received under the various scheduling algorithms with 16 users per sector. The first thing that should be noticed is the increase in the proportion of packets for all algorithms at the highest rate of 2457.6 kb/s. The better estimation of the channel leads to users realizing they can support that highest rate more often. A similar effect happens for 1843.2 kb/s, particularly in the ITU pedestrian B channel; in the ITU pedestrian A channel, the increase is nowhere near as pronounced. Corresponding to the increase at the upper two rates, there is a drop in the proportion of packets sent at 1228.8 kb/s. Packets that were being sent at that rate previously instead are sent at the two higher rates thanks to the better prediction. Perhaps the most interesting observation is the increase in the number of packets being sent at 921.6 kb/s. This is the highest rate format for which transmission occurs over more than one slot in the pedestrian channels. When using margins, the fact that transmission would occur over two slots made that format somewhat unfavorable. A comparatively high margin was required to obtain a 1% packet error rate with that format, lending the users to instead request the lower 614.4 kb/s format instead. With perfect prediction, the trouble with the change in the SIR level over the two slots is removed, and so that format can be selected more often. Corresponding in part to the increase at 921.6 kb/s is a decrease at 614.4 kb/s. As mentioned above, since users can request the higher rate format of 921.6 kb/s with more accuracy, this results in a decrease in those users requesting 614.4 kb/s. Also coupled into the decrease is the fact that perfect prediction tends to eliminate early terminations. Therefore, packets that may have terminated early using margins instead request that higher rate directly in perfect prediction. We see the increases in the proportions of rates below 614.4 kb/s, which correspond to those lower rate packets that no longer end early. Those packets may have achieved a rate of 614.4 kb/s before thanks to the margins, but instead are now distributed over the set of lower rates.

The lack of early termination is also reflected in Figures C.7 and C.8 in Appendix C, which show the distributions of formats and slots used for packets transmitted under the various scheduling algorithms with 16 users per sector. Notice that there are very few packets received using less than half of the maximum number of re-transmission attempts using perfect prediction. When using margins, most of the early terminations occurred after half or less of the maximum number of re-transmission attempts for the lower rate packets. Therefore, the effective bit rates of those lower rate formats are smaller under perfect prediction than when using margins.

Figures C.9 and C.10 in Appendix C show the average delay per packet per drop as a function of the number of users per sector for the various scheduling algorithms using perfect multislot prediction. For all the algorithms except the maxD algorithm, there is an increase in the average delay seen; the maxD delay remains almost the same as in the case of margins. The amount of increase in the delay again is related to how strongly the algorithm emphasizes the requested rate. Those algorithms that place the least emphasis on channel conditions and transmit the most often to low rate users show the largest jumps in delay. The RR, EXP1A and EXP2A delay increases are particularly noticeable. By transmitting to the users who request low rates more often, it is possible for the delay of those packets to in fact increase when compared to if the system were to wait for a while for the supportable rates of those users to improve. More prominent, however, is the fact that the packets of other users experience longer delays, first by not being served in the first place instead of the lower rate users, and then by reducing the chance of any new packet being transmitted for a time as slots are reserved for re-transmission of the low rate packets. Figure C.11 and Figure C.12 in Appendix C show the average packet delay per user per drop as a function of the number of users per sector. These curves also increase, and for the same reasons as the average delay per packet increased. There is one exception, however: the maxD algorithm in the ITU pedestrian A channel showed a decrease in the delay per user. This is related to the change in the probability of no service. Since in the pedestrian A channel, there is a reduced chance of receiving no service, the average delay per user also decreases when moving to perfect prediction. In the pedestrian B channel, however, the probability of no service increases, and hence so does the delay per user.

Figure C.13 to Figure C.16 in Appendix C show the distributions of packet delays for the various scheduling algorithms with 16 users per sector. There is generally an overall reduction in the probability of lengthy packet delays for all algorithms when compared to the case with margins. Correspondingly, the probability of very small packet delays is also reduced. Since low data rate packets are sent more often with perfect prediction, their delay (which tends to be the longest of all packets) is reduced, causing a somewhat increased delay for the high data rate packets. The following tables compare the percentiles and standard deviations of packet delay for the various scheduling algorithms with perfect multislot prediction in the two pedestrian channels.

TABLE XLIX:
PERCENTILES AND STANDARD DEVIATIONS OF PACKET DELAYS FOR SCHEDULING ALGORITHMS IN THE ITU PEDESTRIAN A CHANNEL WITH 16 USERS PER SECTOR COMPARING USE OF MARGINS WITH PERFECT MULTISLOT PREDICTION

Value	Using Margins / Using Perfect Prediction					
Algorithm	RR	maxD	PF	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75
50th percentile	25.9 / 41.1	4 / 4	4 / 4.2	4 / 4	4 / 4	4 / 4
90th	45.2 / 64.8	8.9 / 9.5	20.9 / 33.3	10.1 / 12.1	12.4 / 16.4	15.5 / 23.5
95th	58.9 / 75.9	17.3 / 18.7	90.4 / 104.8	57.9 / 62.2	69.5 / 78.7	80.5 / 94.5
99th	123.4 / 129.3	164.7 / 164.3	322.6 / 342.1	372.7 / 388.4	359.6 / 378.0	340.9 / 360.5
99.5th	177.2 / 179.6	329.3 / 324.0	448.3 / 464.1	606.6 / 616.6	537.3 / 552.0	485.0 / 500.6
99.9th	438.9 / 414.4	1476.4 / 1437.2	837.9 / 814.8	1245.8 / 1227.2	1037.0 / 1021.9	922.2 / 896.8
99.95th	655.1 / 603.5	2835.4 / 2833.7	1095.7 / 1014.3	1607.8 / 1534.6	1338.0 / 1258.1	1190.6 / 1108.1

99.99th	1870.2 / 1549.2	8673.6 / 8516.4	2345.1 / 1668.3	2783.6 / 2297.4	2516.8 / 1988.1	2442.6 / 1816.6
Std. Dev.	52.06 / 45.54	168.80 / 167.33	80.64 / 73.75	98.76 / 92.16	90.11 / 83.08	84.42 / 77.49
Algorithm	M-LWDF 1	M-LWDF 2	slot-fair	packet-fair	minperf 5 kb/s	minperf 10 kb/s
50th percentile	12.3 / 14.1	15.8 / 21.1	4 / 4	4 / 4.1	4 / 4	4 / 4.2
90th	38.6 / 45.5	41.5 / 55.6	12.4 / 19.4	29.8 / 57.8	10.1 / 11.7	11.5 / 15.5
95th	59.6 / 67.5	59.2 / 73.5	86.0 / 101.4	111.7 / 140.0	29.8 / 36.9	39.9 / 55.5
99th	132.3 / 144.7	121.2 / 131.0	372.9 / 399.0	371.7 / 445.0	351.1 / 413.8	462.3 / 577.9
99.5th	184.8 / 195.5	163.0 / 165.4	535.6 / 563.0	500.0 / 604.9	849.8 / 939.9	782.9 / 865.4
99.9th	419.6 / 380.0	373.0 / 332.8	1017.6 / 1022.3	842.9 / 1053.7	1746.7 / 1682.6	1453.1 / 1453.1
99.95th	596.1 / 504.7	549.8 / 469.8	1293.2 / 1243.3	1019.6 / 1274.1	2129.0 / 2016.1	1759.2 / 1726.4
99.99th	1579.1 / 1121.7	1479.7 / 1243.0	2180.6 / 1966.3	1785.9 / 2041.9	3097.6 / 2859.7	2610.6 / 2455.6
Std. Dev.	48.53 / 40.56	47.54 / 40.30	87.73 / 85.30	81.53 / 94.00	120.51 / 118.95	109.00 / 112.98
Algorithm	minperf 15 kb/s	minperf 20 kb/s	minperf 25 kb/s	EXP1A	EXP2A	EXP1B
50th percentile	4 / 4.3	4 / 4.5	4 / 4.5	21.5 / 34.0	22.9 / 36.7	4.8 / 5.0
90th	12.5 / 20.7	15.0 / 27.3	15.6 / 28.9	47.7 / 64.4	46.4 / 65.7	7.6 / 7.9
95th	50.7 / 66.7	60.3 / 81.4	68.4 / 93.0	62.6 / 78.7	60.0 / 78.9	25.3 / 28.5
99th	490.2 / 608.8	492.9 / 647.9	498.1 / 678.1	115.7 / 129.5	112.6 / 128.9	362.1 / 371.3
99.5th	733.0 / 850.8	716.3 / 890.9	711.0 / 949.1	156.9 / 173.4	157.5 / 174.2	735.0 / 709.9
99.9th	1307.5 / 1416.3	1226.4 / 1480.7	1201.1 / 1509.3	364.8 / 384.7	382.7 / 393.0	1632.6 / 1494.0
99.95th	1572.1 / 1670.8	1483.6 / 1703.9	1427.9 / 1765.0	531.9 / 560.3	552.2 / 564.8	2047.3 / 1932.3
99.99th	2332.4 / 2351.7	2210.9 / 2433.6	2115.9 / 2445.6	1341.4 / 1445.7	1471.6 / 1500.9	3571.8 / 3390.3
Std. Dev.	103.51 / 115.59	101.98 / 123.04	102.31 / 128.63	42.34 / 43.97	44.62 / 44.27	116.60 / 109.50
Algorithm	EXP2B	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75		
50th percentile	4.2 / 4.6	4 / 4	4 / 4	4 / 4.3		
90th	40.9 / 51.7	21.5 / 25.5	28.3 / 36.3	34.9 / 45.4		
95th	100.7 / 111.8	78.9 / 86.9	90.7 / 101.3	97.2 / 108.3		
99th	274.6 / 282.4	328.2 / 333.0	302.6 / 308.1	287.2 / 293.6		
99.5th	361.5 / 363.7	444.0 / 442.9	405.8 / 404.2	382.6 / 381.2		
99.9th	666.3 / 594.8	793.3 / 740.1	732.0 / 672.6	697.3 / 626.7		
99.95th	894.4 / 722.4	1018.1 / 887.5	975.5 / 807.0	930.0 / 757.4		
99.99th	1909.3 / 1203.7	2068.3 / 1400.4	2034.6 / 1293.6	1927.4 / 1255.8		
Std. Dev.	69.71 / 61.08	77.23 / 68.17	74.50 / 64.58	71.13 / 62.61		

TABLE L:
PERCENTILES AND STANDARD DEVIATIONS OF PACKET DELAYS FOR SCHEDULING ALGORITHMS IN THE ITU PEDESTRIAN B CHANNEL WITH 16 USERS PER SECTOR COMPARING USE OF MARGINS WITH PERFECT MULTISLOT PREDICTION

Value	Using Margins / Using Perfect Prediction					
Algorithm	RR	maxD	PF	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75
50th percentile	30.7 / 39.7	4 / 4	4 / 4	4 / 4	4 / 4	4 / 4
90th	45.1 / 59.6	9.9 / 9.4	19.6 / 35.9	8.0 / 11.1	11.0 / 14.5	13.6 / 21.1
95th	55.4 / 69.0	51.1 / 46.9	113.0 / 130.7	74.8 / 87.2	92.8 / 110.1	105.2 / 122.6
99th	105.8 / 100.3	263.2 / 258.4	353.6 / 378.8	395.9 / 421.7	385.5 / 410.4	370.4 / 395.0
99.5th	148.4 / 136.8	449.6 / 444.9	461.3 / 484.4	576.7 / 599.4	527.8 / 549.5	491.9 / 514.6
99.9th	344.4 / 297.0	1399.0 / 1442.1	777.2 / 755.6	1085.6 / 1075.9	937.6 / 919.0	843.0 / 821.9
99.95th	493.2 / 421.5	2245.7 / 2279.3	998.5 / 894.9	1362.2 / 1308.8	1162.2 / 1096.3	1069.3 / 986.4
99.99th	1216.4 / 1013.8	5549.0 / 5782.8	1987.0 / 1310.6	2348.8 / 1926.5	2170.0 / 1634.0	2087.2 / 1439.7
Std. Dev.	30.10 / 31.48	132.28 / 137.08	76.93 / 74.18	90.09 / 87.70	83.5 / 81.42	79.93 / 77.52
Algorithm	M-LWDF 1	M-LWDF 2	slot-fair	packet-fair	minperf 5 kb/s	minperf 10 kb/s
50th percentile	18.2 / 21.5	20.9 / 27.3	4 / 4	4 / 4	4 / 4	4 / 4
90th	41.9 / 51.2	44.3 / 56.6	15.9 / 21.7	46.5 / 60.4	12.0 / 12.9	15.6 / 19.4
95th	61.5 / 71.5	59.9 / 71.7	116.8 / 131.8	140.4 / 164.9	62.2 / 62.1	75.0 / 81.1
99th	120.5 / 138.4	110.2 / 118.4	399.2 / 424.9	395.4 / 466.3	389.6 / 455.0	456.7 / 548.3
99.5th	165.0 / 179.6	145.0 / 144.5	520.3 / 548.9	504.9 / 610.6	691.7 / 789.9	685.1 / 759.5
99.9th	361.5 / 322.6	317.0 / 265.9	813.4 / 832.7	761.5 / 953.3	1446.1 / 1413.1	1183.6 / 1230.4
99.95th	521.5 / 406.7	447.7 / 370.1	949.1 / 954.6	881.5 / 1119.5	1709.0 / 1677.4	1404.3 / 1443.7
99.99th	1223.4 / 828.2	1103.2 / 846.0	1382.2 / 1258.8	1333.4 / 1496.8	2335.5 / 2281.4	1917.2 / 1938.9
Std. Dev.	37.78 / 33.53	34.15 / 31.19	78.16 / 80.80	77.95 / 91.60	100.27 / 104.07	94.03 / 101.96

Algorithm	minperf 15 kb/s	minperf 20 kb/s	minperf 25 kb/s	EXP1A	EXP2A	EXP1B
50th percentile	4 / 4	4 / 4	4 / 4	27.7 / 36.2	29.3 / 37.6	4.2 / 4
90th	19.6 / 27.2	21.7 / 29.0	19.9 / 27.6	47.6 / 59.6	47.6 / 60.4	11.7 / 13.4
95th	93.2 / 102.0	105.4 / 116.3	103.5 / 113.7	59.4 / 71.9	58.2 / 71.7	64.0 / 65.0
99th	485.6 / 579.1	511.1 / 607.6	542.1 / 657.4	101.0 / 107.9	99.1 / 105.8	365.7 / 369.1
99.5th	680.8 / 787.8	701.6 / 855.6	757.8 / 941.4	137.1 / 137.2	136.5 / 136.3	590.6 / 569.5
99.9th	1112.3 / 1272.8	1152.9 / 1370.1	1257.3 / 1477.2	294.7 / 286.6	297.0 / 290.8	1127.6 / 1251.3
99.95th	1288.0 / 1491.4	1327.2 / 1580.7	1435.5 / 1669.4	411.1 / 405.6	422.1 / 408.3	1502.2 / 1612.8
99.99th	1759.0 / 2017.3	1749.2 / 2111.8	1885.3 / 2115.8	958.6 / 932.9	1014.2 / 943.3	2650.7 / 2618.6
Std. Dev.	94.37 / 108.93	98.57 / 117.33	105.47 / 125.04	31.37 / 31.12	31.86 / 30.99	92.25 / 92.17

Algorithm	EXP2B	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75
50th percentile	4 / 4.4	4 / 4	4 / 4	4 / 4
90th	51.5 / 64.7	25.7 / 38.3	36.2 / 51.9	45.1 / 59.3
95th	122.9 / 137.8	110.3 / 123.0	118.5 / 132.6	121.5 / 136.5
99th	283.4 / 290.9	322.7 / 326.7	303.9 / 310.9	293.2 / 300.7
99.5th	356.1 / 356.4	417.4 / 420.8	388.2 / 390.6	371.0 / 371.7
99.9th	609.4 / 527.0	707.0 / 660.9	656.5 / 595.4	633.0 / 557.4
99.95th	820.7 / 612.5	917.4 / 773.4	874.3 / 693.4	850.6 / 649.3
99.99th	1709.0 / 883.4	1820.7 / 1111.7	1760.7 / 1009.3	1722.3 / 950.1
Std. Dev.	65.93 / 60.12	71.06 / 65.45	67.98 / 63.10	67.23 / 61.56

Figures C.17 and C.18 in Appendix C show the distributions of average packet delay per user per drop for the various scheduling algorithms with 16 users per sector. As with the average delay per user curves for varying numbers of users discussed early, there is again a shift to higher delays for almost all of the distributions. The exception again is the maxD algorithm, where the delays appear to shift to smaller values. Recall that these graphs filter out the users who do not receive any service for an entire drop, whereas the averages versus the number of users per sector do not. Therefore, it can be said that for the users who do actually receive data under the maxD algorithm, their average delays are smaller with perfect prediction than when using margins. Take into account those starved users, however, and the delay would likely increase in the pedestrian B channel, as seen earlier. The following tables compare the percentiles and standard deviations of delays per user in the two pedestrian channels.

TABLE LI:
PERCENTILES AND STANDARD DEVIATIONS OF AVERAGE PACKET DELAY PER USER FOR SCHEDULING ALGORITHMS IN THE ITU PEDESTRIAN A CHANNEL WITH 16 USERS PER SECTOR COMPARING USE OF MARGINS WITH PERFECT MULTISLOT PREDICTION

Value	Using Margins / Using Perfect Prediction					
Algorithm	RR	maxD	PF	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75
10th percentile	24.8 / 37.0	7.5 / 7.6	15.9 / 16.7	10.4 / 10.8	12.3 / 12.8	14.1 / 14.9
25th	26.6 / 40.0	13.4 / 13.2	17.0 / 18.0	14.0 / 14.0	15.1 / 15.4	16.0 / 16.7
50th	30.3 / 44.7	39.3 / 38.1	19.1 / 20.5	26.3 / 25.4	21.6 / 21.9	19.8 / 20.7
75th	37.8 / 51.8	186.4 / 163.3	21.8 / 25.0	46.1 / 47.0	29.9 / 32.2	24.5 / 27.2
90th	58.9 / 67.3	725.9 / 605.2	30.9 / 35.3	71.1 / 77.1	44.5 / 49.1	35.9 / 39.7
Std. Dev.	26.4 / 19.7	1102.1 / 1170.5	14.1 / 14.0	40.7 / 39.7	24.3 / 23.2	17.8 / 17.1
Algorithm	M-LWDF 1	M-LWDF 2	slot-fair	packet-fair	minperf 5 kb/s	minperf 10 kb/s
10th percentile	12.0 / 13.5	18.0 / 22.5	17.0 / 17.5	19.6 / 22.5	8.5 / 8.8	9.3 / 10.2
25th	15.9 / 17.7	19.6 / 24.8	17.6 / 17.9	20.2 / 24.1	16.8 / 17.0	18.6 / 20.4
50th	25.3 / 27.5	22.7 / 29.0	18.9 / 19.7	21.2 / 26.8	56.7 / 60.5	44.2 / 53.8
75th	43.7 / 45.3	28.1 / 36.1	21.6 / 24.9	23.2 / 30.6	83.1 / 100.4	54.8 / 66.3
90th	86.2 / 84.0	42.9 / 51.9	30.8 / 42.6	27.7 / 39.5	97.7 / 117.1	64.8 / 76.1
Std. Dev.	61.1 / 47.9	20.7 / 18.5	50.2 / 27.6	11.7 / 11.2	38.5 / 43.9	22.8 / 25.8
Algorithm	minperf 15 kb/s	minperf 20 kb/s	minperf 25 kb/s	EXP1A	EXP2A	EXP1B

10th percentile	10.3 / 12.1	12.1 / 15.7	14.2 / 19.9	18.3 / 28.3	22.4 / 32.7	8.0 / 8.1
25th	21.0 / 25.4	23.3 / 31.0	23.4 / 31.2	22.2 / 33.2	24.2 / 36.0	15.4 / 14.9
50th	36.2 / 45.2	30.8 / 39.8	27.4 / 36.1	29.3 / 40.4	27.4 / 41.3	62.9 / 57.1
75th	43.1 / 52.6	36.4 / 45.3	32.4 / 41.5	40.7 / 51.5	34.0 / 49.8	437.3 / 365.6
90th	50.8 / 59.7	42.6 / 51.6	38.0 / 47.1	63.4 / 72.1	50.1 / 67.0	1482.0 / 1280.1
Std. Dev.	16.3 / 18.1	12.2 / 13.6	10.1 / 10.7	28.7 / 24.5	21.4 / 20.5	1070.1 / 707.2

Algorithm	EXP2B	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75
10th percentile	16.1 / 16.9	10.5 / 11.0	12.5 / 13.0	14.3 / 15.0
25th	17.3 / 18.2	14.3 / 14.2	15.3 / 15.6	16.2 / 16.9
50th	19.4 / 20.7	26.4 / 25.8	21.9 / 22.2	20.1 / 21.0
75th	22.3 / 25.4	46.2 / 46.5	30.5 / 32.5	25.0 / 27.5
90th	31.7 / 35.5	70.1 / 76.0	43.9 / 49.3	35.9 / 40.2
Std. Dev.	13.9 / 14.1	36.4 / 38.8	22.2 / 23.2	16.6 / 17.3

TABLE LII:
PERCENTILES AND STANDARD DEVIATIONS OF AVERAGE PACKET DELAY PER USER FOR SCHEDULING ALGORITHMS IN THE ITU PEDESTRIAN B CHANNEL WITH 16 USERS PER SECTOR COMPARING USE OF MARGINS WITH PERFECT MULTISLOT PREDICTION

Value	Using Margins / Using Perfect Prediction					
Algorithm	RR	maxD	PF	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75
10th percentile	28.4 / 35.7	9.1 / 9.0	17.7 / 18.8	11.8 / 11.9	13.8 / 14.3	15.9 / 16.7
25th	30.2 / 38.3	13.4 / 13.1	18.7 / 19.9	14.9 / 15.5	16.2 / 17.1	17.3 / 18.6
50th	33.0 / 43.0	31.3 / 30.8	20.4 / 23.0	23.0 / 24.8	21.5 / 23.5	20.6 / 22.9
75th	39.0 / 49.0	102.7 / 98.0	23.2 / 26.7	38.2 / 41.8	28.3 / 32.4	24.7 / 28.5
90th	52.7 / 58.9	337.9 / 277.1	32.0 / 34.2	56.4 / 62.8	40.5 / 45.1	34.8 / 37.6
Std. Dev.	17.7 / 13.1	710.5 / 902.6	21.9 / 10.7	32.9 / 28.4	25.5 / 17.6	23.2 / 13.2
Algorithm	M-LWDF 1	M-LWDF 2	slot-fair	packet-fair	minperf 5 kb/s	minperf 10 kb/s
10th percentile	16.0 / 18.6	22.1 / 27.3	17.9 / 18.0	21.8 / 24.1	10.1 / 10.2	11.2 / 11.8
25th	18.5 / 21.6	22.7 / 28.8	18.8 / 19.2	22.7 / 26.0	14.8 / 15.4	16.6 / 18.5
50th	25.7 / 29.8	25.4 / 31.8	20.8 / 22.9	24.1 / 28.9	35.7 / 38.8	36.2 / 42.1
75th	39.5 / 45.1	31.3 / 38.5	23.8 / 27.6	26.5 / 32.8	70.4 / 84.2	48.4 / 57.1
90th	68.8 / 74.7	43.7 / 51.4	34.1 / 41.6	31.0 / 39.3	81.2 / 97.2	53.5 / 62.5
Std. Dev.	59.9 / 31.8	15.5 / 14.0	15.2 / 18.5	9.48 / 8.14	31.6 / 34.8	19.5 / 20.4
Algorithm	minperf 15 kb/s	minperf 20 kb/s	minperf 25 kb/s	EXP1A	EXP2A	EXP1B
10th percentile	13.0 / 14.1	16.1 / 18.4	20.8 / 24.0	24.6 / 31.7	27.3 / 33.9	9.3 / 9.3
25th	19.8 / 23.5	24.1 / 29.0	25.4 / 29.6	26.6 / 34.5	28.8 / 36.5	13.8 / 13.9
50th	34.3 / 40.0	30.4 / 35.6	27.8 / 32.7	31.1 / 39.7	31.7 / 40.9	34.3 / 35.3
75th	38.6 / 45.1	33.4 / 39.5	30.7 / 36.8	39.9 / 48.6	38.0 / 47.8	118.3 / 128.2
90th	42.1 / 49.2	36.5 / 43.4	33.8 / 41.3	54.8 / 61.6	49.9 / 58.9	472.2 / 500.6
Std. Dev.	14.7 / 13.6	10.8 / 9.76	9.11 / 7.57	19.0 / 15.9	16.1 / 13.9	375.2 / 374.4

Algorithm	EXP2B	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75
10th percentile	18.1 / 19.1	12.0 / 12.1	14.1 / 14.5	16.2 / 16.9
25th	19.0 / 20.3	15.2 / 15.9	16.6 / 17.4	17.8 / 18.9
50th	20.9 / 23.3	23.4 / 25.1	22.1 / 23.8	21.1 / 23.2
75th	23.6 / 27.0	38.4 / 41.7	28.8 / 32.5	25.2 / 28.8
90th	32.0 / 34.7	56.5 / 62.7	40.6 / 45.8	35.0 / 38.0
Std. Dev.	18.5 / 10.8	31.2 / 28.4	24.5 / 17.7	22.5 / 13.2

It can be seen that although in most cases the percentiles of delay per user increase in value, the standard deviation of those delays tends to decrease. Therefore, using perfect multislot prediction treats the set of users requesting service somewhat more fairly.

Figure C.19 and Figure C.20 in Appendix C show the distributions of allocated slots per user per drop for the various scheduling algorithms with 16 users per sector. It is a bit more difficult to make generalizations about the slot distributions than it was regarding the other distributions. It seems that most of the algorithms display a narrowing of the distribution when moving to perfect multislot prediction, but there also seem to be some exceptions. The following tables provide a bit more insight.

TABLE LIII:
PERCENTILES AND STANDARD DEVIATIONS OF ALLOCATED SLOTS PER USER PER DROP FOR SCHEDULING ALGORITHMS IN THE ITU PEDESTRIAN A CHANNEL WITH 16 USERS PER SECTOR COMPARING USE OF MARGINS WITH PERFECT MULTISLOT PREDICTION

Value	Using Margins / Using Perfect Prediction					
Algorithm	RR	maxD	PF	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75
10th percentile	840 / 581	0 / 0	766 / 941	289 / 355	473 / 590	617 / 785
25th	940 / 715	22 / 28	1040 / 1036	451 / 470	718 / 737	907 / 916
50th	1130 / 1026	275 / 301	1163 / 1125	775 / 810	976 / 978	1101 / 1071
75th	1312 / 1460	1433 / 1491	1290 / 1218	1623 / 1624	1487 / 1450	1387 / 1326
90th	1436 / 1832	3628 / 3562	1401 / 1307	2419 / 2280	1945 / 1823	1615 / 1512
Std. Dev.	247.7 / 474.0	1794.6 / 1730.1	265.9 / 151.7	926.0 / 834.5	592.5 / 501.1	392.8 / 287.0
Algorithm	M-LWDF 1	M-LWDF 2	slot-fair	packet-fair	minperf 5 kb/s	minperf 10 kb/s
10th percentile	383 / 599	942 / 936	990 / 1011	961 / 712	210 / 213	324 / 323
25th	636 / 760	1104 / 982	1096 / 1053	1012 / 796	264 / 284	406 / 429
50th	999 / 1030	1161 / 1041	1150 / 1125	1068 / 903	434 / 533	609 / 680
75th	1533 / 1434	1217 / 1197	1232 / 1204	1157 / 1086	1297 / 1339	1168 / 1437
90th	2072 / 1817	1266 / 1463	1281 / 1237	1406 / 1922	3278 / 3064	2841 / 2491
Std. Dev.	646.4 / 463.5	175.3 / 219.6	183.7 / 105.9	242.6 / 632.6	1499.7 / 1355.9	1250.5 / 1054.8
Algorithm	minperf 15 kb/s	minperf 20 kb/s	minperf 25 kb/s	EXP1A	EXP2A	EXP1B
10th percentile	419 / 408	493 / 455	546 / 478	938 / 749	902 / 638	14 / 22
25th	519 / 516	609 / 567	681 / 595	1024 / 843	976 / 757	43 / 52
50th	737 / 759	858 / 806	933 / 822	1096 / 1027	1104 / 1001	292 / 328
75th	1243 / 1482	1375 / 1382	1417 / 1345	1188 / 1337	1266 / 1419	1449 / 1496
90th	2470 / 2406	2040 / 2311	1939 / 2309	1325 / 1655	1402 / 1824	3589 / 3480
Std. Dev.	1007.8 / 877.0	782.0 / 805.7	621.2 / 809.0	249.0 / 350.3	201.0 / 446.6	1759.5 / 1687.5
Algorithm	EXP2B	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75		
10th percentile	792 / 963	324 / 382	514 / 620	667 / 808		
25th	1044 / 1042	473 / 488	733 / 749	914 / 923		
50th	1167 / 1122	785 / 813	983 / 981	1100 / 1070		
75th	1281 / 1208	1604 / 1602	1469 / 1435	1371 / 1311		
90th	1385 / 1292	2393 / 2254	1903 / 1801	1584 / 1492		
Std. Dev.	256.1 / 138.7	897.7 / 811.8	565.8 / 481.1	367.0 / 270.9		

TABLE LIV:
PERCENTILES AND STANDARD DEVIATIONS OF ALLOCATED SLOTS PER USER PER DROP FOR SCHEDULING ALGORITHMS IN THE ITU PEDESTRIAN B CHANNEL WITH 16 USERS PER SECTOR COMPARING USE OF MARGINS WITH PERFECT MULTISLOT PREDICTION

Value	Using Margins / Using Perfect Prediction					
Algorithm	RR	maxD	PF	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75
10th percentile	752 / 666	21 / 20	797 / 996	386 / 435	578 / 667	698 / 849
25th	833 / 758	140 / 128	1075 / 1068	590 / 623	818 / 832	971 / 972
50th	1078 / 952	597 / 574	1190 / 1143	973 / 969	1093 / 1085	1175 / 1142
75th	1379 / 1365	1764 / 1746	1258 / 1189	1574 / 1504	1425 / 1369	1326 / 1268
90th	1571 / 1871	3085 / 3083	1328 / 1222	2077 / 2049	1715 / 1632	1457 / 1358
Std. Dev.	324.4 / 477.5	1280.3 / 1305.4	231.9 / 98.2	678.2 / 633.3	444.1 / 369.2	303.5 / 198.4
Algorithm	M-LWDF 1	M-LWDF 2	slot-fair	packet-fair	minperf 5 kb/s	minperf 10 kb/s
10th percentile	565 / 728	1072 / 944	1050 / 1055	890 / 718	305 / 303	458 / 445
25th	820 / 838	1116 / 992	1096 / 1081	955 / 805	390 / 396	574 / 551
50th	1103 / 1034	1141 / 1047	1142 / 1123	1028 / 913	654 / 697	817 / 832

75th	1443 / 1374	1173 / 1168	1182 / 1163	1159 / 1130	1576 / 1535	1424 / 1469
90th	1691 / 1659	1210 / 1452	1217 / 1197	1583 / 1872	2704 / 2627	2296 / 2223
Std. Dev.	428.8 / 357.1	114.1 / 208.5	113.8 / 67.8	298.9 / 585.3	1028.3 / 1001.5	786.6 / 756.7

Algorithm	minperf 15 kb/s	minperf 20 kb/s	minperf 25 kb/s	EXP1A	EXP2A	EXP1B
10th percentile	578 / 530	647 / 568	673 / 586	925 / 775	825 / 706	43 / 46
25th	676 / 637	751 / 680	769 / 685	981 / 862	893 / 801	167 / 168
50th	930 / 877	968 / 878	960 / 884	1080 / 978	1058 / 953	612 / 595
75th	1416 / 1410	1379 / 1341	1320 / 1268	1245 / 1273	1323 / 1325	1763 / 1740
90th	1948 / 2091	1872 / 2004	1836 / 2022	1401 / 1715	1536 / 1829	3043 / 3018
Std. Dev.	582.6 / 657.2	497.6 / 663.5	501.5 / 686.1	188.1 / 372.4	276.3 / 440.5	1251.5 / 1260.3

Algorithm	EXP2B	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75
10th percentile	849 / 1011	414 / 456	606 / 693	742 / 874
25th	1077 / 1072	609 / 634	826 / 838	979 / 977
50th	1182 / 1139	983 / 972	1093 / 1084	1166 / 1140
75th	1242 / 1182	1552 / 1484	1415 / 1361	1313 / 1259
90th	1304 / 1212	2046 / 2016	1677 / 1610	1432 / 1345
Std. Dev.	206.8 / 88.5	657.4 / 615.2	420.6 / 353.5	279.1 / 185.8

A look at the change in standard deviations confirms that most of the slot distributions become tighter when perfect prediction is used. This is particularly true for the PF, slot-fair, and EXP2B algorithms. The algorithms for which the distributions became more spread out seem to be the same algorithms that showed a decrease in average sector throughput. In other words, the algorithms that do not emphasize channel conditions show a wider spread of allocated slots when using perfect multislot prediction. The two key exceptions to this statement appear to be the maxD and EXP1B algorithms in the ITU pedestrian B channel. In those two cases, the large gain in throughput due to perfect prediction also caused a wider variation in the number of allocated slots across the set of users in the sector.

Overall, it was seen that the throughput increased for some algorithms, while for others, the throughput decreased. The direction and magnitude of throughput change was directly related to how strongly the algorithm metric emphasized requested rates. For a strong emphasis, there was a noticeable increase in throughput; for little or no emphasis on requested rates (e.g. for delays being more important), there was a sharp drop in throughput. The increases were particularly noticeable in the ITU pedestrian B channel, while the decreases were more prominent in the ITU pedestrian A channel, with both effects due to the relative magnitude of signal strength variations in the two channels. The tightness of slot distributions also reflected the change in throughput. Those algorithms that showed a higher throughput also showed a narrower slot distribution, and vice versa. Considering delays, both delays per packet and delays per user increased for almost all algorithms in both channels. However, the tightness of the delay distributions also tended to increase, meaning delays were distributed somewhat more fairly across the set of users requesting service.

Perhaps one of the most important observations is that although throughputs increased and decreased, and some distributions became narrower or more spread apart, the relative positions of the algorithms to each other really did not change when switching from margins to perfect multislot prediction. That is, the algorithms that provided the highest, lowest, or intermediate throughputs and delays were the same using both prediction methods. The relative tightness of the throughput, delay, and slot distributions also were unchanged. Therefore, it can be said that the use of margins with outdated channel information to determine the rates each user can support is an effective alternative to attempting to predict what the signal strength will be in future slots.

5.4.2. "PF A" and "PF B" Average Rate Calculation Comparisons

It was seen in the section on margin usage that for the proportionally fair algorithm, there were no significant changes in any of the results between the two described methods of calculating the average rate. It was postulated that any algorithm that involved the average rate as a multiplication or division would also not show any significant change. Because the minimum performance algorithm involved a subtraction rather than a division, it was analyzed separately, and a noticeable difference in the throughput results was seen.

It is reasonable to assume that the same trend should hold when using perfect multislot prediction. Fortunately, with the use of perfect prediction, the iteration process for the margins is removed. It therefore takes little effort or processing time to run extra simulations. Hence, all algorithms involving the calculation of the average rate were run using the "PF B" calculation method to compare with the results using the "PF A" method.

Figures D.1 and D.2 in Appendix D show a comparison of the average sector throughput per drop as a function of the number of users per sector for the two calculation methods in the two pedestrian channels. It can be seen that for most of the algorithms, there is virtually no difference in the throughput achieved between the "PF A" and "PF B" calculation methods, as expected. However, for the minperf algorithm, there is a significant increase in the sector throughput with the "PF B" method, as there was in the case where margins were used. This once again is a result of the more accurate average rate value provided by the "PF B" method.

Figure D.3 and Figure D.4 in Appendix D show the distributions of throughput per user per drop for the two calculation methods with 16 users per sector. Once again, in most cases, there is no real difference to be seen in the distributions. However, the minperf algorithm shows the same kind of shift in the distributions that was seen when using margins. The sudden jump in the throughput occurs closer to the minimum specified throughput, and the probability of users with large throughputs increases. The following tables compare the statistics and probability of not obtaining the minimum throughput for the two calculation methods.

Table LV:
Statistics for Average Throughput per User per Drop Distributions and Probability of Not Achieving Minimum Specified Throughput for the Minimum Performance Scheduling Algorithm in the ITU Pedestrian A Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate and Perfect Multislot Prediction

Calc. Method	"PF A"					"PF B"				
minperf C_k (kb/s)	5	10	15	20	25	5	10	15	20	25
Mean (kb/s)	128.6	108.1	89.7	73.9	63.3	130.4	112.5	95.5	79.5	67.6
Median (kb/s)	35.0	33.7	32.5	33.1	36.2	36.4	34.1	33.5	33.9	36.1
Std. Dev. (kb/s)	191.9	153.5	119.5	88.2	63.9	195.0	160.4	128.1	97.4	71.8
P[Throughput $<C_k$] (%)	0.81	1.44	1.75	2.81	3.63	0.94	1.56	2.44	3.00	3.88

Table LVI:
Statistics for Average Throughput per User per Drop Distributions and Probability of Not Achieving Minimum Specified Throughput for the Minimum Performance Scheduling Algorithm in the ITU Pedestrian B Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate and Perfect Multislot Prediction

Calc. Method	"PF A"					"PF B"				
minperf C_k (kb/s)	5	10	15	20	25	5	10	15	20	25
Mean (kb/s)	85.9	73.3	60.9	51.4	45.5	87.2	76.5	65.2	55.1	48.1
Median (kb/s)	38.7	34.5	31.4	30.5	33.5	40.3	36.1	33.2	31.4	33.3
Std. Dev. (kb/s)	101.2	81.0	61.4	44.4	30.1	103.0	85.4	67.1	50.0	35.0
P[Throughput $<C_k$] (%)	0.25	0.44	1.06	1.63	3.19	0.31	0.94	1.81	2.69	4.44

As in the case of margins, the probability of not obtaining the minimum specified throughput increases when using the "PF B" method. This is again a consequence of the users in poor channel conditions operating closer to that minimum compared to the "PF A" method.

Figures D.5 and D.6 in Appendix D show the distributions of effective bit rates of packets comparing the two calculation methods in the two pedestrian channels with 16 users per sector. There is yet again basically no change to be seen in the proportions when comparing the two methods for most algorithms in the ITU pedestrian A channel. The ITU pedestrian B channel shows some very minor changes for most of the algorithms. The change in proportions is a lot more pronounced for the minperf algorithm, particularly for larger values of the minimum specified throughput. As in the case of margins, a larger proportion of the packets are received at the highest bit rates, while fewer packets are received at lower bit rates.

Figures D.7 and D.8 in Appendix D show the distributions of formats and number of slots used to correctly receive packets with 16 users per sector. The only difference seen between the two calculation methods is again for the minperf algorithm. The same shift can be seen towards higher rate formats as was seen for the effective bit rates.

Figures D.9 and D.10 in Appendix D show the average delay per packet per drop as a function of the number of users per sector, comparing the two calculation methods. There is yet to be seen any significant change in the curves other than for the minperf algorithm. Even for the minperf algorithm, although there is a decrease in the packet delay, that decrease is fairly minor. The same statements can be said regarding the average packet delay per user per drop curves seen in Figures D.11 and D.12 in Appendix D. The only difference is that the delay values increase instead of decrease. The change is even smaller than for the delay per packet, and for larger values of the minimum specified throughput, the change essentially becomes insignificant.

Figures D.13 through D.16 in Appendix D show a comparison of the distributions of packet delays for the two calculation methods. As above, there still is no significant difference in most of the distributions. For the minperf algorithm, there is a shift that increases in size for larger values of the minimum specified throughput towards a decreased probability of large packet delays. This same trend was seen when using margins as well, and the overall shift in both cases is still fairly small. The following tables compare the percentiles and standard deviations for the packet delay distributions for the minperf algorithm using both calculation methods.

Table LVII:
Percentiles and Standard Deviations of Packet Delays for the Minimum Performance Scheduling Algorithm in the ITU Pedestrian A Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate and Perfect Multislot Prediction

Calc. Method	"PF A"					"PF B"				
C_k (kb/s) / Percentile	5	10	15	20	25	5	10	15	20	25
50th (median)	4	4.2	4.3	4.5	4.5	4	4.1	4.2	4.4	4.4
90th	11.7	15.5	20.7	27.3	28.9	11.5	15.0	19.5	26.6	28.8
95th	36.9	55.5	66.7	81.4	93.0	36.0	53.2	65.3	83.3	97.1
99th	413.8	577.9	608.8	647.9	678.1	408.8	556.7	580.0	600.8	633.9
99.5th	939.9	865.4	850.8	890.9	949.1	919.5	840.4	804.0	821.2	878.6
99.9th	1682.6	1453.1	1416.3	1480.7	1509.3	1656.9	1410.2	1346.5	1383.7	1439.9
99.95th	2016.1	1726.4	1670.8	1703.9	1765.0	1973.3	1702.4	1608.4	1606.0	1675.4
99.99th	2859.7	2455.6	2351.7	2433.6	2445.6	2914.8	2408.5	2211.5	2297.6	2338.8
Std. Dev.	118.95	112.98	115.59	123.04	128.63	117.51	109.92	109.81	115.12	122.10

Table LVIII:
Percentiles and Standard Deviations of Packet Delays for the Minimum Performance Scheduling Algorithm in the ITU Pedestrian B Channel with 16 Users per Sector Using the "PF A" and "PF B" Methods to Calculate the Average Rate and Perfect Multislot Prediction

Calc. Method	"PF A"					"PF B"				
C_k (kb/s) / Percentile	5	10	15	20	25	5	10	15	20	25
50th (median)	4	4	4	4	4	4	4	4	4	4
90th	12.9	19.4	27.2	29.0	27.6	12.5	17.2	25.1	29.8	28.6
95th	62.1	81.1	102.0	116.3	113.7	61.4	79.2	103.3	120.9	124.9
99th	455.0	548.3	579.1	607.6	657.4	443.5	523.7	540.9	561.0	599.4
99.5th	789.9	759.5	787.8	855.6	941.4	782.9	730.4	723.9	768.3	861.2
99.9th	1413.1	1230.4	1272.8	1370.1	1477.2	1406.0	1182.8	1173.3	1266.5	1389.9

99.95th	1677.4	1443.7	1491.4	1580.7	1669.4	1674.2	1402.6	1383.6	1456.9	1566.6
99.99th	2281.4	1938.9	2017.3	2111.8	2115.8	2276.7	1890.2	1838.4	1934.7	2024.5
Std. Dev.	104.07	101.96	108.93	117.33	125.04	103.14	98.07	101.56	108.60	117.40

Note that the standard deviations decrease for the "PF B" method, which was also seen in the case of margins.

Figures D.17 and D.18 in Appendix D show the distributions of average packet delay per user per drop comparing the "PF A" and "PF B" calculation methods with 16 users per sector. The minperf algorithm again shows the only visible change. As in the case with the use of margins, there is a general widening of the distributions, with low delays becoming shorter and high delays becoming longer. The following table compares the percentiles and standard deviations of the delays per user. Note that the standard deviations increase from the "PF A" to "PF B" method, confirming how the distributions become more spread apart.

TABLE LIX:
PERCENTILES AND STANDARD DEVIATION OF AVERAGE PACKET DELAY PER USER FOR THE MINIMUM PERFORMANCE SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN A CHANNEL WITH 16 USERS PER SECTOR USING THE "PF A" AND "PF B" METHODS TO CALCULATE THE AVERAGE RATE AND PERFECT MULTISLOT PREDICTION

Calc. Method	"PF A"					"PF B"				
C_k (kb/s) Percentile	5	10	15	20	25	5	10	15	20	25
10th	8.8	10.2	12.1	15.7	19.9	8.8	9.9	11.6	14.0	17.8
25th	17.0	20.4	25.4	31.0	31.2	16.8	19.3	23.2	28.7	31.2
50th (median)	60.5	53.8	45.2	39.8	36.1	58.5	55.4	46.4	40.1	36.5
75th	100.4	66.3	52.6	45.3	41.5	104.1	67.7	53.5	46.0	41.4
90th	117.1	76.1	59.7	51.6	47.1	119.8	76.6	60.6	51.6	46.5
Std. Dev.	43.9	25.8	18.1	13.6	10.7	45.6	26.8	18.9	14.3	11.2

TABLE LX:
PERCENTILES AND STANDARD DEVIATION OF AVERAGE PACKET DELAY PER USER FOR THE MINIMUM PERFORMANCE SCHEDULING ALGORITHM IN THE ITU PEDESTRIAN B CHANNEL WITH 16 USERS PER SECTOR USING THE "PF A" AND "PF B" METHODS TO CALCULATE THE AVERAGE RATE AND PERFECT MULTISLOT PREDICTION

Calc. Method	"PF A"					"PF B"				
C_k (kb/s) Percentile	5	10	15	20	25	5	10	15	20	25
10th	10.2	11.8	14.1	18.4	24.0	10.0	11.3	13.2	16.6	21.9
25th	15.4	18.5	23.5	29.0	29.6	15.1	17.4	21.6	26.8	29.7
50th (median)	38.8	42.1	40.0	35.6	32.7	37.7	40.2	40.3	36.1	32.7
75th	84.2	57.1	45.1	39.5	36.8	85.8	59.2	46.3	39.6	36.4
90th	97.2	62.5	49.2	43.4	41.3	101.5	65.2	50.4	44.1	40.2
Std. Dev.	34.8	20.4	13.6	9.76	7.57	36.7	21.8	14.8	10.6	7.86

Figures D.19 and D.20 in Appendix D show the distributions of allocated slots per user per drop for the two calculation methods with 16 users per sector. Unlike the case with margins, there is not a very large change to be seen in the distributions for any of the algorithms, including the minperf algorithm. Nevertheless, the difference does exist, and is analyzed by comparing the percentiles and standard deviations in the tables below.

TABLE LXI:
PERCENTILES AND STANDARD DEVIATION OF SLOTS ALLOCATED PER USER PER DROP FOR SCHEDULING ALGORITHMS IN THE ITU PEDESTRIAN A CHANNEL WITH 16 USERS PER SECTOR USING THE "PF A" AND "PF B" METHODS TO CALCULATE THE AVERAGE RATE AND PERFECT MULTISLOT PREDICTION

Calc. Method	"PF A"			"PF B"		
Algorithm	PF	maxD/PF 0.25	maxD/PF 0.5	PF	maxD/PF 0.25	maxD/PF 0.5
10th percentile	941	355	590	913	333	570
25th	1036	470	737	1037	467	736
50th	1125	810	978	1132	809	983
75th	1218	1624	1450	1228	1634	1455
90th	1307	2280	1823	1318	2291	1835
Std. Dev.	151.7	834.5	501.1	167.2	845.6	513.2
Algorithm	maxD/PF 0.75	M-LWDF 2	minperf 5 kb/s	maxD/PF 0.75	M-LWDF 2	minperf 5 kb/s
10th percentile	785	936	213	751	948	209
25th	916	982	284	917	992	279
50th	1071	1041	533	1077	1050	518
75th	1326	1197	1339	1335	1197	1326
90th	1512	1463	3064	1523	1438	3077
Std. Dev.	287.0	219.6	1355.9	300.8	204.2	1381.6
Algorithm	minperf 10 kb/s	minperf 15 kb/s	minperf 20 kb/s	minperf 10 kb/s	minperf 15 kb/s	minperf 20 kb/s
10th percentile	323	408	455	321	414	465
25th	429	516	567	419	518	574
50th	680	759	806	670	771	823
75th	1437	1482	1382	1403	1442	1412
90th	2491	2406	2311	2566	2367	2266
Std. Dev.	1054.8	877.0	805.7	1094.6	887.3	774.0
Algorithm	minperf 25 kb/s	EXP2A	EXP2B	minperf 25 kb/s	EXP2A	EXP2B
10th percentile	478	638	963	500	646	939
25th	595	757	1042	612	763	1042
50th	822	1001	1122	839	1010	1128
75th	1345	1419	1208	1357	1411	1217
90th	2309	1824	1292	2261	1808	1302
Std. Dev.	809.0	446.6	138.7	749.1	438.6	151.1
Algorithm	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75
10th percentile	382	620	808	368	600	791
25th	488	749	923	487	748	924
50th	813	981	1070	817	984	1075
75th	1602	1435	1311	1608	1439	1319
90th	2254	1801	1492	2258	1809	1502
Std. Dev.	811.8	481.1	270.9	819.4	489.9	281.9

TABLE LXII:
PERCENTILES AND STANDARD DEVIATION OF SLOTS ALLOCATED PER USER PER DROP FOR SCHEDULING ALGORITHMS IN THE ITU PEDESTRIAN B CHANNEL WITH 16 USERS PER SECTOR USING THE "PF A" AND "PF B" METHODS TO CALCULATE THE AVERAGE RATE AND PERFECT MULTISLOT PREDICTION

Calc. Method	"PF A"			"PF B"		
Algorithm	PF	maxD/PF 0.25	maxD/PF 0.5	PF	maxD/PF 0.25	maxD/PF 0.5
10th percentile	996	435	667	967	422	650
25th	1068	623	832	1063	620	831
50th	1143	969	1085	1150	968	1089
75th	1189	1504	1369	1199	1515	1379
90th	1222	2049	1632	1235	2063	1647
Std. Dev.	98.2	633.3	369.2	115.5	645.2	382.6
Algorithm	maxD/PF 0.75	M-LWDF 2	minperf 5 kb/s	maxD/PF 0.75	M-LWDF 2	minperf 5 kb/s

10th percentile	849	944	303	823	961	293
25th	972	992	396	974	1005	385
50th	1142	1047	697	1149	1056	681
75th	1268	1168	1535	1276	1161	1553
90th	1358	1452	2627	1373	1422	2671
Std. Dev.	198.4	208.5	1001.5	214.5	187.5	1027.1
Algorithm	minperf 10 kb/s	minperf 15 kb/s	minperf 20 kb/s	minperf 10 kb/s	minperf 15 kb/s	minperf 20 kb/s
10th percentile	445	530	568	440	543	582
25th	551	637	680	540	638	689
50th	832	877	878	816	886	914
75th	1469	1410	1341	1483	1413	1355
90th	2223	2091	2004	2268	2101	1958
Std. Dev.	756.7	657.2	663.5	788.0	634.6	604.8
Algorithm	minperf 25 kb/s	EXP2A	EXP2B	minperf 25 kb/s	EXP2A	EXP2B
10th percentile	586	706	1011	610	713	992
25th	685	801	1072	710	809	1066
50th	884	953	1139	909	959	1144
75th	1268	1325	1182	1312	1320	1190
90th	2022	1829	1212	1951	1817	1224
Std. Dev.	686.1	440.5	88.5	613.9	430.7	101.4
Algorithm	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75
10th percentile	456	693	874	447	673	850
25th	634	838	977	630	837	972
50th	972	1084	1140	970	1087	1145
75th	1484	1361	1259	1491	1369	1266
90th	2016	1610	1345	2030	1624	1356
Std. Dev.	615.2	353.5	185.8	623.0	363.2	197.9

From the tables, it can be seen that the distributions of slots tend to widen when switching to the "PF B" calculation method, although the difference is rather small. The exceptions to this statement seem to be the same algorithms that showed a decrease in throughput when switching from margins to perfect prediction.

To summarize, using the "PF B" method to calculate the average rate did not significantly change the results for any of the algorithms except for the minperf algorithm. For the minperf algorithm, the throughput increased, the distributions became wider, the probability of not meeting the minimum specified throughput increased, and the average delay per packet and per user only changed by a small amount. All these effects were similar to what occurred when the switch from "PF A" to "PF B" was examined for the minperf algorithm while using margins. This therefore shows again that outdated channel information and margins can effectively replace prediction, and that the "PF B" method only improves the results for the minperf algorithm in the pedestrian channels.

5.5. ITU Vehicular B Channel, Outdated SIR$_p$s Plus Margins

5.5.1. Results from the ITU vehicular B channel

As discussed in Chapter 4, the handling of the ITU vehicular B channel is different from that of the pedestrian channel analyzed earlier in two key ways. Due to the higher mobile speed and subsequently faster fading, there is not much correlation between the values of the SIR three slots apart. Hence, it does not seem to make much sense to attempt to use the current SIR value (even with a margin) in order to request a format for a transmission that would occur three slots after the SIR estimation is made. Instead, the format determination is made based on the current SIR_p value, which is the current SIR value with the Rayleigh fading component removed. In other words, the small-scale fading is essentially averaged out when determining the supportable transmission format to request from the base station. Note however that when determining if the packet is received correctly or not, the Rayleigh component is still included. Secondly, in order to add a measure of time diversity, for those rates where more than one format is available, the format that takes the larger number of slots is selected for the vehicular channel. This is the opposite from the pedestrian channels, where the format with fewer slots was chosen. To be more specific, for the bit rates of 307.2 kb/s, 614.4 kb/s, and 1228.8 kb/s, whereas formats DRC4, DRC5, and DRC8 would have been used respectively in the pedestrian channels, DRC6, DRC7, and DRC11 are instead used respectively for the vehicular channel.

Using these two changes, Figure E.1 in Appendix E shows the average sector throughput per drop as a function of the number of users per sector for the ITU vehicular B channel. The first observation is that the resulting throughputs are clearly much lower than in the pedestrian channels. The throughput gain is also much lower. For example, in the pedestrian channels, the throughput for the maxD algorithm increased from 1 to 32 users by a factor of approximately 3. In the ITU vehicular B channel, the gain is only a factor of about 2.2. Other algorithms are even worse, with the gain flattening out rather quickly. Many algorithms show almost no throughput gain at all (i.e. the PF, EXP2B, slot-fair, and M-LWDF 2 algorithms), and some even show a drop in throughput as the number of users in the sector increases. These effects are primarily due to the loss of an accurate measure of the current SIR level when determining the supportable rate for each user.

There are a few algorithms that are particularly noteworthy. First, the relative throughput of the M-LWDF 1 algorithm to the others has increased. The delay factor in the metric must in a fashion help to ameliorate the throughput loss due to the poor channel estimate. The same statement can be made regarding the EXP1A and EXP2A algorithms, which both display throughputs comparable to that seen in the ITU pedestrian B channel. Secondly, notice that the throughput of the packet-fair algorithm is in fact lower than that of the RR algorithm. Clearly, to take advantage of the channel conditions when dividing the packets evenly amongst users, the estimate of the channel conditions must be reliable. Otherwise, the decision about who to schedule when is not particularly useful, and just choosing a user at random or in cyclical order can in fact be better. The division of the throughput range between the maxD and PF throughputs with the tuning parameter in the maxD/PF algorithm is not quite as linear as in the pedestrian channels. However, there is still a good separation between the resulting curves.

Perhaps the most interesting curves belong to the minperf algorithm. The initial gain in throughput due to multiuser diversity stops after only a few users per sector. This is as high as 8 users per sector for a minimum specified throughput of 5 kb/s, but is as low as 2 users for 25 kb/s. The poor channel estimation effectively causes a reduction in the capacity of the channel, meaning it is more difficult to meet the minimum specification. Beyond the above mentioned number of users, the throughput decreases similar to what was seen for the pedestrian channels, as providing the minimum throughput to all users becomes the driving goal of the algorithm. The curves then reach a certain minimum throughput. At this point, it becomes infeasible to provide the minimum throughput to all the users requesting service. Instead, that minimum is only provided to the users with medium to very good signal strengths, while the users with poor signal strengths are largely ignored. In extreme cases, this can mean that certain users can in fact go without service for an entire drop. Of particular note are the 20 kb/s and 25 kb/s cases with 32 users per sector. The resulting throughput achieved by providing the minimum throughput to a reduced set of users becomes larger than the throughput for small numbers of users per sector when all users received at least that minimum.

Figure E.2 in Appendix E shows the distributions of throughputs per user per drop with 16 users per sector in the ITU vehicular B channel. The throughputs distributions cover a much smaller range than the pedestrian channels. That is, the throughput achieved by each user is much smaller in the vehicular channel than pedestrian channels, which reflects the results from the average sector throughput. Furthermore, there seems to be a higher proportion of users with comparatively low throughputs than in the pedestrian channels; the distributions tend to increase more sharply than in the pedestrian channels. There is a much larger probability of users not receiving any service, not only for the maxD algorithm, but for the minperf algorithm as well. The probability of a user not receiving any service for the maxD algorithm and the minperf algorithm for increasing values of minimum specified throughput are respectively 53.81%, 0.19%, 0.31%, 0.44%, 1.19%, and 1.75%.

The minperf distributions are again interesting to look at. In comparison to the pedestrian channels, there is a much larger proportion of the users operating around the minimum throughput in the vehicular channel. As the minimum specified throughput increases, the tightness of the distribution around that minimum tends to decrease, however. This is the opposite of what happens in the pedestrian channels. There is also an increasing proportion of users who receive very little throughput (e.g. below 10 kb/s). The increase in spread and low throughput users is a symptom of how users in poor channel conditions are increasingly ignored in the vehicular B channel under the minperf algorithm due to a lack of capacity from poor channel estimates. The probability of not achieving the minimum specified throughput for the five tested values of minimums in increasing order are respectively 1.63%, 2.75%, 6.31%, 10.13%, and 14.75%. This is a large increase from the percentages seen for the pedestrian channels.

The following table compares the statistics of the distributions.

TABLE LXIII:
MEAN, MEDIAN, AND STANDARD DEVIATION OF AVERAGE THROUGHPUT PER USER PER DROP DISTRIBUTIONS FOR SCHEDULING ALGORITHMS IN THE ITU VEHICULAR B CHANNEL WITH 16 USERS PER SECTOR

Algorithm	RR	maxD	PF	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75
Mean (kb/s)	28.4	72.2	39.4	61.5	52.1	44.6
Median (kb/s)	29.5	0	38.5	16.7	29.5	37.2
Std. Dev. (kb/s)	12.8	181.3	22.9	113.7	65.6	35.6

Algorithm	M-LWDF 1	M-LWDF 2	slot-fair	packet-fair	minperf 5 kb/s	minperf 10 kb/s
Mean (kb/s)	45.5	33.2	36.7	27.1	37.0	25.8
Median (kb/s)	34.0	33.5	35.5	27.0	14.3	24.6
Std. Dev. (kb/s)	43.0	17.3	21.9	11.1	64.4	13.2

Algorithm	minperf 15 kb/s	minperf 20 kb/s	minperf 25 kb/s	EXP1A	EXP2A	EXP1B
Mean (kb/s)	30.1	34.6	38.2	36.1	30.5	71.5
Median (kb/s)	31.2	36.4	41.1	30.3	30.9	1.0
Std. Dev. (kb/s)	8.3	10.8	13.5	30.8	14.8	178.3

Algorithm	EXP2B	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75
Mean (kb/s)	38.9	60.2	51.1	43.8
Median (kb/s)	38.3	17.7	29.7	36.9
Std. Dev. (kb/s)	22.4	109.1	62.8	34.4

Figure E.3 in Appendix E shows the distributions of effective bit rates and Figure E.4 shows the distributions of formats and number of slots used for transmitted packets in the ITU vehicular B channel with 16 users per sector. These figures help show why the throughput is dramatically lower in the vehicular channel. Note that no packets were sent using the three highest rate formats. This is because it turned out that the DRC10, DRC11, and DRC12 formats could not be supported at a 1% packet error rate while using the SIR_p values for format determination. Initially, with the margins set at zero, the packet error rates for those formats were extremely high (e.g. about 75% for DRC12). As the margins increased, the formats effectively became cut off before the 1% PER was achieved. That is, the margins became sufficiently large before a 1% PER was reached such that the three formats in question could subsequently not be chosen. Note that the rates of 1228.8 kb/s and 1843.2 kb/s were still obtained, but this was entirely due to early terminations of packets being sent with DRC7 and DRC9. There is a question of if these high packet error rates for the three formats make sense. For the answer, consider the following graph.

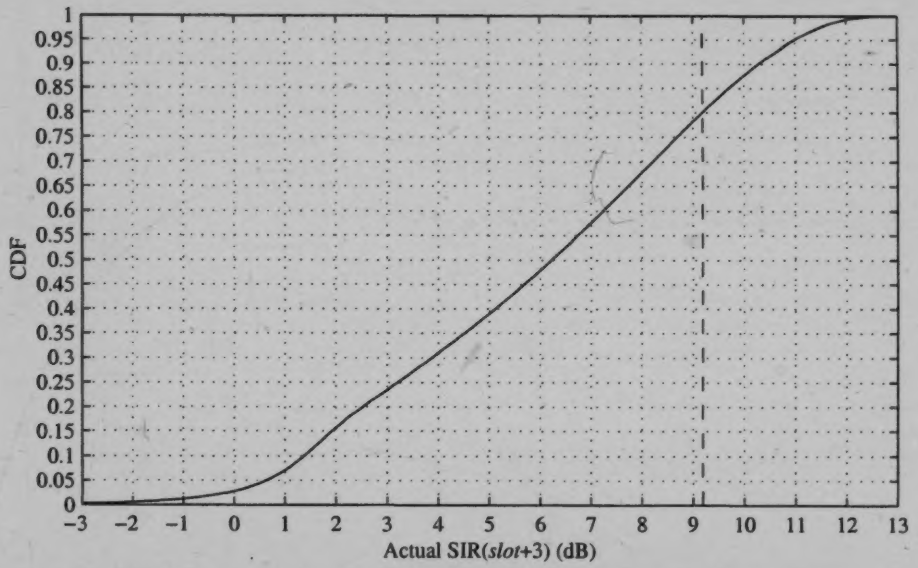


Figure 5.187: CDF of actual SIR 3 slots in the future (including Rayleigh component and imperfections) for the ITU vehicular B channel, given that the SIR_p (with imperfections, no Rayleigh component) in the current slot supports DRC12 (i.e. SIR_p>9.187 dB)

From the graph, it can be seen that if the current SIR_p level is above the threshold required for DRC12 (i.e. the signal-to-noise ratio required for a 1% PER in the AWGN channel), only about 20% of the time will the actual SIR still be above that threshold 3 slots later. Since it is possible for the packet to be decoded correctly for an SIR below that threshold (e.g. about 0.5 dB below the threshold), albeit with a rapidly increasing likelihood of error, this means that when the current SIR_p is greater than 9.187 dB, only about 25% of the time will the packet be decoded correctly. Thus, the 75% initial packet error rate with a margin of 0 dB does indeed make sense. Furthermore, in order to obtain a 1% PER, from the graph, the margin would have to be about 10 or 11 dB. However, as soon as a margin of 3.9 dB is used (13 dB – 9.187 dB, rounded up), the format of DRC12 would be cut off. With a margin of 3.9 dB, the user would require a current SIR_p level of 3.9 + 9.187 = 13.087 dB to request DRC12. However, since 13 dB is the maximum possible SIR or SIR_p value when imperfections are taken into account, DRC12 would therefore never be selected. The explanation for DRC10 and DRC11 is similar.

As also discussed earlier, there is obviously a much larger proportion of packets received at lower bit rates. The proportions of packets received at each rate still appear to be about the same for the maxD, PF, and maxD/PF algorithms compared to their EXP and maxD/PF/EXP counterparts. The change in proportions with the tuning parameter is not quite linear anymore, however. The linear trend with the minimum specified throughput for the minperf algorithm also no longer exists, due to the ignoring nature the algorithm develops towards users in poor channel conditions at high values of the minimum throughput.

Figure E.5 in Appendix E shows the average delay per packet per drop as a function of the number of users per sector for the ITU vehicular B channel. Clearly, the delay for all the algorithms is larger compared to the pedestrian channels. Note however that the delay of the M-LWDF 1 algorithm is relatively lower compared to the other algorithms. For example, in the pedestrian channels, the M-LWDF 1 algorithm delay was larger than that of the PF algorithm, whereas in the vehicular channel, the M-LWDF 1 delay is smaller. This reflects how the throughput of the M-LWDF 1 algorithm was also relatively larger than the PF algorithm as seen earlier. Note also that the delay of the packet-fair algorithm is larger than that of the RR algorithm, just as the throughput of the packet-fair algorithm was smaller than that of the RR algorithm.

For the minperf algorithm, the delays initially are larger for increasing values of the minimum specified throughput up to about 4 users per sector. Then, similar to how the sector throughputs curve upwards, the rate of the packet delays curves downwards. At 32 users per sector, the average delay per packet is actually smaller for larger values of the minimum throughput. The effect is similar to what normally occurs for the maxD algorithm, although on a smaller scale. With certain users in the sector effectively being ignored under the minperf algorithm, the few packets of those users experience very long delays. In comparison, the packets of the smaller group of users that are served more often experience shorter delays. Therefore, as with the maxD algorithm, there is a large group of packets with small delays, and a small group with large delays, leading to an overall small value for the average delay per packet. However, one can therefore expect the average delay per user in those cases to be rather large. This will be examined shortly.

Figure E.6 in Appendix E shows the average packet delay per user per drop as a function of the number of users per sector for the vehicular channel. As with the delay per packet, the delay per user is substantially larger compared to the values obtained in the pedestrian channels. There is a noticeable increase in the delay seen by the maxD/PF and maxD/PF/EXP algorithms, an increase which is more conspicuous the smaller the value of the tuning parameter. The increase in the delay for the maxD algorithm is particularly striking, and is primarily due to the large proportion of users who do not receive any service during a given drop.

Just as significant as the delay of the maxD algorithm is the delay seen under the minperf algorithm. As predicted above, the delay per user is extremely large. Since the algorithm starts to ignore certain users due to the lack of capacity, those users see very large delays and in some cases receive no service during a drop. Thus, the average delay per user is extremely high. As expected from the delay per packet, the delay per user becomes larger the higher the value of the minimum specified throughput is. The other algorithms are not particularly noteworthy, and, other than the larger values, do not act much differently than in the pedestrian channels.

Figures E.7 and E.8 in Appendix E show the distributions of delay per packet for the vehicular channel with 16 users per sector. As already seen, the delays are larger than those experienced in the pedestrian channels. There is a significant reduction in the delay under the EXP1B, EXP2B, and maxD/PF/EXP algorithms when compared to the delays of the maxD, PF, and maxD/PF algorithms, respectively. This reduction in delay was also seen in the pedestrian channels, but the amount of reduction is greater in the vehicular channel. Note that the reduction in delay is not quite as apparent under the maxD/EXP1B comparison as in the others. Remember that the packets which are not transmitted at all during a drop are not included in the maxD distribution, whereas those same packets are served under the EXP1B algorithm, and consequently included in the distribution. There is also the fact that the EXP1B algorithm must quite often allocate service away from the user who requests the highest rate in the vehicular channel in order to reduce the wait of long-delayed packets of other users. Thus, the delay of the highest user is increased more that it is in the pedestrian channels, which in turn creates a shift to the right in the EXP1B distribution relative to the maxD distribution. However, for delays larger than about 4500 slots, the EXP1B algorithm does provide a lower probability of those large delays than the maxD algorithm.

For the minperf algorithm, there is an increasing likelihood of large packet delays as the specified minimum throughput increases. As the minimum increases, it is more likely that a packet from a user in poor channel conditions will be ignored for a fairly long time to satisfy the minimum throughput of better users. Thus, the distribution increasingly shifts towards longer delays.

The following table compares the percentiles and standard deviations of the packet delay distributions.

TABLE LXIV:
PERCENTILES AND STANDARD DEVIATIONS OF PACKET DELAYS FOR SCHEDULING ALGORITHMS IN THE ITU VEHICULAR B CHANNEL WITH 16 USERS PER SECTOR

Algorithm	RR	maxD	PF	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75
50th percentile	39.6	7.3	16.6	7.7	9.8	13.4
90th	55.2	12.2	57.5	42.5	47.8	52.4
95th	62.5	15.1	90.0	89.5	83.2	87.0
99th	86.8	26.5	451.5	439.9	435.9	451.8
99.5th	129.6	55.1	698.2	926.0	872.3	792.9
99.9th	846.8	1304.6	1425.3	1949.3	1710.4	1546.8
99.95th	1504.7	3033.7	1838.5	2420.0	2118.1	1923.5
99.99th	4094.2	9444.9	3907.1	3969.3	3778.4	3829.8
Std. Dev.	92.54	179.05	121.65	142.38	132.36	125.54
Algorithm	M-LWDF 1	M-LWDF 2	slot-fair	packet-fair	minperf 5 kb/s	minperf 10 kb/s
50th percentile	21.7	33.0	20.2	27.5	8.0	8.0
90th	52.8	58.1	70.0	74.5	28.0	25.3
95th	89.1	72.9	114.1	126.0	59.2	44.9
99th	170.3	113.1	358.0	466.3	966.5	1229.0
99.5th	226.8	140.2	480.3	658.3	1320.4	1841.1
99.9th	680.5	759.0	912.5	1337.5	2384.4	3184.3
99.95th	1269.0	1413.1	1404.0	1874.0	3026.9	3822.5
99.99th	3670.8	3946.2	3870.3	4342.8	4731.5	5937.9
Std. Dev.	86.72	89.90	105.06	126.71	197.96	249.42
Algorithm	minperf 15 kb/s	minperf 20 kb/s	minperf 25 kb/s	EXP1A	EXP2A	EXP1B
50th percentile	7.7	7.5	7.4	34.6	37.7	7.6
90th	17.0	15.4	13.5	58.2	56.5	11.8
95th	26.8	19.7	16.8	70.6	65.6	13.5

99th	1069.3	587.2	349.0	103.2	92.6	586.9
99.5th	2160.3	2073.4	1881.8	127.5	118.2	1398.4
99.9th	3799.0	4157.6	4404.2	724.0	785.4	3111.0
99.95th	4612.6	5007.6	5414.8	1378.9	1434.6	3919.0
99.99th	7141.0	8074.2	8866.1	3886.2	4043.6	5780.2
Std. Dev.	274.52	281.92	287.69	88.36	90.59	204.05

Algorithm	EXP2B	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75
50th percentile	19.4	8.3	11.7	16.0
90th	61.6	61.3	55.4	59.0
95th	100.0	102.6	98.8	97.9
99th	352.8	437.7	400.1	375.7
99.5th	452.8	576.1	520.8	483.9
99.9th	943.8	1096.4	1020.2	966.3
99.95th	1438.4	1459.6	1412.5	1408.0
99.99th	3810.1	3491.7	3571.3	3680.0
Std. Dev.	104.22	109.52	105.74	104.27

Notice that the median delay per packet is no longer 4 slots, even for algorithms such as maxD which allocate many packets to a single user. Part of this is due to the fact that the ITU vehicular B channel varies quite rapidly, which allows a larger group of users to request the highest rate more often. The other part is that the highest data rates that only take one slot to transmit are no longer used because they can't be supported at a 1% packet error rate. Instead, formats that nominally take two slots to transmit a packet are used, although there are early terminations, so that some packets are received after only one slot. The combination of the two factors leads to a median packet delay greater than 4 slots.

Figure E.9 in Appendix E shows the distributions of the average packet delay per user per drop for the vehicular channel with 16 users per sector. Upon first glance, it would appear that that the EXP1B algorithm displays the largest delays per user, even larger than the maxD algorithm. However, it must be remembered that the 53.8% of the users that do not receive any service during an entire drop are filtered out of the maxD distribution. Thus, the distribution for the maxD algorithm is somewhat artificially raised. If the starved users were not filtered out, the distribution would effectively be multiplied by 0.462 (i.e. 1 - 0.538) and would look more like the graph below.

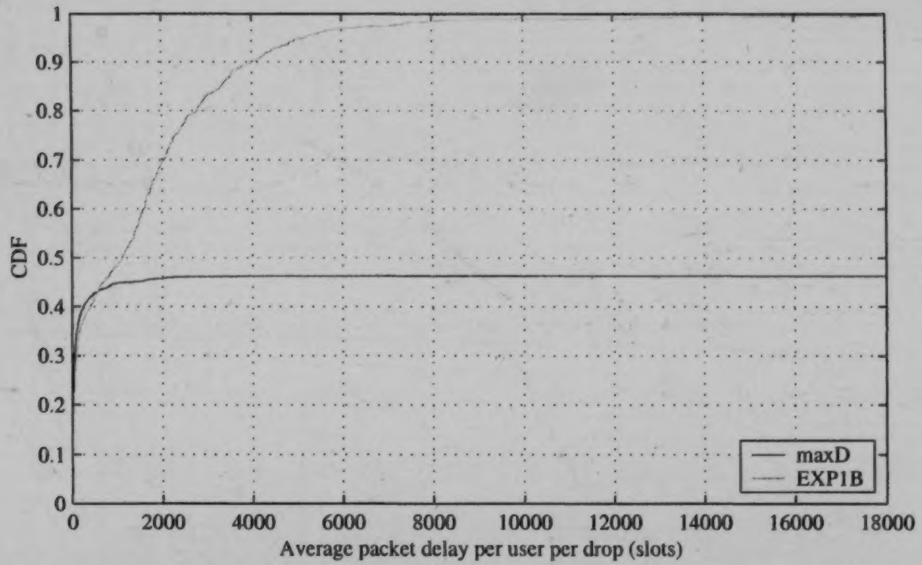


Figure 5.188: Distributions of average packet delay per user for all users (including those whose receive no service for an entire drop) in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins

Because there is no way to know what the actual delay of the starved users would eventually be beyond the end of a drop, they are instead filtered out of the distribution.

For the minperf algorithm, with a minimum specified throughput of 5 kb/s, the delay distribution still shows the same type of two-step increase as was seen in the pedestrian channels, since with the minimum, it is still possible to provide the said minimum to all users in the sector. For larger minimum specifications, it is no longer possible. As such, the step disappears, since there is no longer a certain group receiving around the minimum throughput and another receiving much more than the minimum throughput. Instead most users receive about the minimum throughput, and others receive less than the minimum specified throughput. As the minimum specified throughput increases, the tail at the upper end of the distribution tends to become longer, as more users in the worse channel conditions experience longer delays.

The remaining algorithms, aside from the larger values of delay, show much the same shape for the distributions as in the pedestrian channels. The following table compares the percentiles and standard deviations of the delay per user distributions as shown in Appendix E.

TABLE LXV:
PERCENTILES AND STANDARD DEVIATIONS OF AVERAGE PACKET DELAY PER USER FOR SCHEDULING ALGORITHMS IN THE ITU VEHICULAR B CHANNEL WITH 16 USERS PER SECTOR FOR USERS WHO RECEIVE AT LEAST ONE PACKET PER DROP

Algorithm	RR	maxD	PF	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75
10th percentile	36.5	8.4	25.1	13.4	16.7	20.5
25th	38.6	11.9	27.6	31.0	25.5	25.4
50th	41.8	29.0	35.5	76.6	46.2	37.5

75th	49.2	102.0	47.5	126.0	71.5	54.0
90th	73.8	375.0	82.9	235.8	136.6	99.5
Std. Dev.	56.7	374.3	57.1	160.2	94.8	69.4
Algorithm	M-LWDF 1	M-LWDF 2	slot-fair	packet-fair	minperf 5 kb/s	minperf 10 kb/s
10th percentile	18.6	30.5	26.4	39.3	21.0	34.3
25th	24.6	32.7	28.5	41.9	52.3	39.0
50th	39.6	38.8	37.8	46.5	63.4	45.8
75th	73.2	49.9	53.8	54.1	76.7	53.2
90th	172.3	76.5	93.5	63.2	90.7	63.0
Std. Dev.	185.7	53.7	53.0	33.8	52.4	36.1
Algorithm	minperf 15 kb/s	minperf 20 kb/s	minperf 25 kb/s	EXP1A	EXP2A	EXP1B
10th percentile	29.1	26.3	24.8	26.2	33.4	11.6
25th	33.1	29.8	27.7	33.1	36.3	51.3
50th	38.0	33.9	31.2	41.6	40.7	1118.8
75th	44.9	39.1	36.3	56.4	49.6	2300.7
90th	59.1	52.2	48.3	93.3	74.9	4012.1
Std. Dev.	55.9	48.7	92.7	73.0	50.2	2281.9

Algorithm	EXP2B	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75
10th percentile	25.5	13.9	17.1	20.7
25th	28.1	31.3	25.8	25.9
50th	35.8	72.9	45.9	37.9
75th	48.0	117.5	69.7	54.4
90th	82.9	216.5	133.0	99.2
Std. Dev.	55.1	147.8	95.8	67.5

Figure E.10 in Appendix E shows the distributions of allocated slots per user per drop in the ITU vehicular B channel with 16 users per sector. These distributions lend themselves to a direct comparison between channels more easily than the other results do. In general, it seems that the distributions of slots are more spread apart when compared to the pedestrian channels. The following table provides a numerical comparison of the distributions.

Table LXVI:
Percentiles and Standard Deviations of Allocated Slots per User per Drop for Scheduling Algorithms in the ITU Vehicular B Channel with 16 Users per Sector

Algorithm	RR	maxD	PF	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75
10th percentile	744	0	961	299	536	765
25th	850	0	1086	346	649	912
50th	1076	0	1165	452	829	1068
75th	1374	541	1234	974	1246	1292
90th	1621	4009	1294	2979	2168	1610
Std. Dev.	359.8	2707.5	203.1	1633.8	872.4	385.0
Algorithm	M-LWDF 1	M-LWDF 2	slot-fair	packet-fair	minperf 5 kb/s	minperf 10 kb/s
10th percentile	459	960	1093	666	373	451
25th	728	1022	1113	757	502	634
50th	1071	1120	1139	938	839	913
75th	1444	1250	1173	1333	1496	1466
90th	1867	1373	1200	1971	2091	2173
Std. Dev.	567.5	217.3	130.5	519.3	925.6	679.4
Algorithm	minperf 15 kb/s	minperf 20 kb/s	minperf 25 kb/s	EXP1A	EXP2A	EXP1B
10th percentile	500	568	595	877	831	15
25th	705	778	814	993	937	19
50th	995	1086	1140	1095	1102	30
75th	1430	1369	1387	1214	1315	549
90th	1939	1795	1678	1399	1502	3910
Std. Dev.	579.8	490.5	453.9	347.5	287.6	2660.6

Algorithm	EXP2B	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75
10th percentile	989	342	575	799
25th	1097	389	685	939
50th	1165	487	845	1070
75th	1222	976	1237	1278
90th	1277	2869	2103	1575
Std. Dev.	195.4	1558.7	824.0	361.9

A comparison of the standard deviations confirms that the distributions are generally less narrow in the vehicular channel than they were in the pedestrian channels. The interesting exception is the minperf algorithm, which displays a tighter distribution in the vehicular channel than it did in the pedestrian channels. The most likely reason for this is that although there is a greater chance of a user being ignored for a long period of time, the act of providing the minimum throughput to a reduced set of users means that the upper tail of the distribution is much shorter in the vehicular channel than in the pedestrian channels. That is, in the pedestrian channel there is no one user or a small group of users who experience extremely good channel conditions and consequently are allocated a large number of slots. Instead, because of both the insufficient capacity and the faster variations of the vehicular channel, the users who aren't ignored are treated more equally. This leads to a tighter distribution in the vehicular channel, despite the fact that more users are ignored for a period of time than in the pedestrian channels.

To summarize, there was a general worsening of results in the ITU vehicular B channel compared to the pedestrian channels. The throughputs decreased, the delays increased, and the distributions tended to become more spread out. For algorithms such as maxD and minperf, there was a much higher probability of both not obtaining the minimum specified throughput (if applicable) and not receiving any service at all during the course of a drop. The three highest rate formats could not be supported at a 1% packet error rate. Early terminations were also seen quite often for all formats due to the lack of an accurate channel estimation. It would seem that the small-scale Rayleigh component of the fading signal is extremely important in obtaining multiuser diversity. By averaging it out, or otherwise ignoring it when determining the rate that can be supported, the whole system operated much more poorly, regardless of the scheduling algorithm.

5.5.2. "PF A" vs. "PF B" Performance Comparison

Running the proportionally fair scheduling algorithm in the ITU vehicular B channel using the "PF B" method to calculate the average rate produced a significant change in the all the analyzed results. This was to be expected, as it was seen earlier that a considerable number of the packets experience early terminations in the vehicular channel. Therefore, a calculation that considers those early terminations should show a noticeable change from one that doesn't.

It therefore made sense to simulate all the algorithms that used the average rate calculation to see the differences the two calculation methods created.

Figure F.1 in Appendix F shows a comparison of the average sector throughput per drop for the two calculation methods in the ITU vehicular B channel. The throughput for all the algorithms increases when using the "PF B" method. The more accurate calculation method means that the average rate value is higher than under the "PF A" method. As a result of the more accurate average, for all algorithms except the minperf algorithm, the algorithm will more accurately choose a user with the best channel conditions relative to that average, instead of selecting users who seem to be in good channel conditions thanks a momentary channel fluctuation. Under the "PF A" method, the users may seem to be above their average, but because of the inaccurate calculation, they may not actually be. By choosing a user who truly is above average, the throughput will increase due to selecting a user with a higher requested rate or that is more likely to experience an early termination. In the case of the minperf algorithm, the more accurate calculation method lets users operate closer to the actual minimum specified throughput, a benefit which was described in more detail in previous sections. The resulting gain in throughput for the minperf algorithm from the "PF B" method is the largest of the algorithms analyzed, as it was in the previous sections. The M-LWDF 2 and EXP2A algorithms show the smallest gain as those algorithms place little emphasis on the channel conditions.

Figure F.2 in Appendix F shows a comparison of the distributions of average throughput per user per drop for the two calculation methods in the vehicular channel. Most of the algorithms show a general widening of the distribution under the "PF B" method, with the high throughput users receiving more throughput, and the low throughput users receiving less throughput. This reflects the gain seen in sector throughput; the relative amount of shift around the lower end of the distributions seems approximately equal to the relative amount of gain in sector throughput. Of particular note is the minperf algorithm. The jump in the throughput occurs much closer to the actual minimum specified throughput under the "PF B" method than under the "PF A" method. With more users operating closer to that minimum, system resources are freed up to give to the users in the best channel conditions. Consequently, the user throughputs at the upper end of the distribution increase. This leads to a longer tail at the upper end of the distribution, meaning there is more variability in the throughput each user in the sector receives. (Note that this lengthened tail is actually seen for all the algorithms, not just the minperf algorithm.) The probability of not achieving the minimum throughput also increases, as seen in previous sections. Under the "PF B" method, the probability of not obtaining the minimum is 2.38%, 3.81%, 6.50%, 12.06%, and 17.63% for the five minimum specified throughputs in increasing order. Fortunately, although the probability of not receiving the minimum throughput increases, the likelihood of not receiving any throughput at all decreases, with probabilities of 0.13%, 0.25%, 0.31%, 0.31%, and 0.38%.

The following table compares the statistics of the distributions between the "PF A" and the "PF B" methods.

TABLE LXVII:
MEAN, MEDIAN, AND STANDARD DEVIATION OF AVERAGE THROUGHPUT PER USER PER DROP DISTRIBUTIONS FOR SCHEDULING ALGORITHMS IN THE ITU VEHICULAR B CHANNEL WITH 16 USERS PER SECTOR COMPARING THE "PF A" AND "PF B" METHODS OF CALCULATING THE AVERAGE RATE

Calc. Method	"PF A" / "PF B"				
Algorithm	PF	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75	M-LWDF 2
Mean (kb/s)	39.4 / 42.3	61.5 / 63.9	52.1 / 55.5	44.6 / 47.8	33.2 / 34.5
Median (kb/s)	38.5 / 36.8	16.7 / 12.9	29.5 / 24.1	37.2 / 32.8	33.5 / 33.2
Std. Dev. (kb/s)	22.9 / 30.0	113.7 / 123.8	65.6 / 78.5	35.6 / 45.6	17.3 / 20.1
Algorithm	minperf 5 kb/s	minperf 10 kb/s	minperf 15 kb/s	minperf 20 kb/s	minperf 25 kb/s
Mean (kb/s)	37.0 / 52.9	25.8 / 38.6	30.1 / 29.4	34.6 / 26.8	38.2 / 28.7
Median (kb/s)	14.3 / 7.2	24.6 / 13.1	31.2 / 19.1	36.4 / 24.8	41.1 / 29.6
Std. Dev. (kb/s)	64.4 / 117.9	13.2 / 70.2	8.3 / 34.4	10.8 / 15.3	13.5 / 8.6
Algorithm	EXP2A	EXP2B	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75
Mean (kb/s)	30.5 / 31.2	38.9 / 41.7	60.2 / 62.5	51.1 / 54.3	43.8 / 46.9
Median (kb/s)	30.9 / 30.8	38.3 / 36.6	17.7 / 13.9	29.7 / 24.8	36.9 / 32.8
Std. Dev. (kb/s)	14.8 / 16.3	22.4 / 29.3	109.1 / 118.7	62.8 / 74.7	34.4 / 44.0

From the above table we see that the means and standard deviations of the distributions generally increase, while the medians decrease. The minperf algorithm at 15 kb/s, 20 kb/s, and 25 kb/s minimum specified throughputs is the exception to this trend. In these cases, since users operate closer to the specified minimum, the algorithm attempts to provide that minimum to all the users more often under the "PF B" method. In other words, the algorithm ignores the users in poor channel conditions less frequently when compared to the "PF A" method. As a result, the throughput per sector and most throughputs per user drop relative to the values obtained under the "PF A" method. One last interesting observation is that for the minperf algorithm, the smallest value of the standard deviation does not necessarily occur at the value of minimum throughput that gives the smallest sector throughput (at 16 users per sector, since that is the case being considered for the throughputs per user). Note that the standard deviation for the 25 kb/s case is smaller than for the 20 kb/s case, even though the average sector throughput and mean throughput per user is larger for 25 kb/s. The cause of this is a bit uncertain, but may have something to so with the fact that the slope around the 16 user per sector point seems to be smaller for the 25 kb/s case than it is for the 20 kb/s case.

Figures F.3 and F.4 in Appendix F show the distributions of effective bit rates and distributions of formats and slots used respectively for packets transmitted in the vehicular channel with 16 users per sector while comparing the "PF A" and "PF B" calculation methods. As expected from the above throughput results, the "PF B" method shows a larger proportion of packets received using the higher rates and formats. Just as with the throughput results, the minperf algorithm displays the largest change in the proportions between the "PF A" and "PF B" methods. Note, however, that the three highest rate formats of DRC10, DRC11, and DRC 12 still are not used to transmit packets. The change in the average rate calculation method makes no difference to the fact that a 1% packet error rate for those formats is not possible when using SIR_p values to determine the supportable rate.

Figures F.5 and F.6 in Appendix F respectively show the average delay per packet per drop and the average packet delay per user per drop, with both as a function of the number of users per sector and comparing the "PF A" and "PF B" calculation methods. As seen many times before for the scheduling algorithms, the increase in throughput for the "PF B" method also corresponds to an overall decrease in the average delay per packet and an increase in the average packet delay per user. Generally, the change is also fairly small for 16 users per sector and below. The exception to the general trend is once again the minperf algorithm, which reverses the decrease/increase trend at higher numbers of users per sector. Even this is not particularly surprising, however. Looking back at the average sector throughput, it was seen that there was an increase in throughput relative to the "PF A" method at low numbers of users per sector, since users in poor channel conditions operated close to the minimum and better users received more throughput. At higher numbers of users, the "PF B" throughput was lower compared to the "PF A" method, as those same poorer users were ignored less often. Therefore, the throughput curves crossed over each other between 4 and 16 users per sector. This same kind of crossover is seen for the delay curves of the minperf algorithm, and in the same range of users that the throughput curves crossed over. Hence, the tradeoff between throughput, delay per packet, and delay per user is seen once again.

Figures F.7 and F.8 in Appendix F compare the distributions of delay per packet for the two calculation methods in the vehicular channel with 16 users per sector. For most of the algorithms, the change in the distributions of delay per packet is negligible. For the minperf algorithm, there is a decreased chance of a fairly short delay (e.g. below 500 slots) using the "PF B" method, but also a decreased likelihood of an extremely long delay (e.g. above 2500 slots). Thus, the distributions of packet delays become tighter under the "PF B" method, as also seen in the pedestrian channels.

The following table compares the percentiles and standard deviations of the packet delay distributions when using the two average rate calculation methods. Only the minperf algorithm is shown, since the change in the other algorithms is negligible.

TABLE LXVIII:
PERCENTILES AND STANDARD DEVIATIONS OF PACKET DELAYS FOR THE MINIMUM PERFORMANCE SCHEDULING ALGORITHM IN THE ITU VEHICULAR B CHANNEL WITH 16 USERS PER SECTOR USING THE "PF A" AND "PF B" METHODS TO CALCULATE THE AVERAGE RATE

Calc. Method	"PF A" / "PF B"				
Algorithm	minperf 5 kb/s	minperf 10 kb/s	minperf 15 kb/s	minperf 20 kb/s	minperf 25 kb/s
50th percentile	8.0 / 7.7	8.0 / 8.0	7.7 / 8.5	7.5 / 8.0	7.4 / 7.8
90th	28.0 / 21.0	25.3 / 33.4	17.0 / 35.2	15.4 / 27.7	13.5 / 19.9
95th	59.2 / 47.1	44.9 / 128.5	26.8 / 129.5	19.7 / 59.6	16.8 / 33.4
99th	966.5 / 776.8	1229.0 / 815.2	1069.3 / 994.6	587.2 / 1155.4	349.0 / 1220.6
99.5th	1320.4 / 1131.0	1841.1 / 1183.5	2160.3 / 1463.8	2073.4 / 1739.4	1881.8 / 1970.6
99.9th	2384.4 / 2149.9	3184.3 / 2171.3	3799.0 / 2583.6	4157.6 / 3018.9	4404.2 / 3484.2
99.95th	3026.9 / 2803.2	3822.5 / 2705.2	4612.6 / 3141.9	5007.6 / 3691.8	5414.8 / 4199.1
99.99th	4731.5 / 4296.5	5937.9 / 4595.3	7141.0 / 5076.7	8074.2 / 5943.9	8866.1 / 6554.9
Std. Dev. (kb/s)	197.96 / 172.18	249.42 / 180.66	274.52 / 209.22	281.92 / 238.59	287.69 / 261.02

Figure F.9 in Appendix F shows the distributions of average packet delay per user per drop comparing the "PF A" and "PF B" methods in the vehicular channel with 16 users per sector. The differences in the curves are clearly more significant than the changes in the packet delay distributions were. There is a general shift towards higher delays per user at the upper end of the distributions. As always, the minperf algorithm displays the greatest change when moving to the "PF B" algorithm. The two-step increase in the distributions is more prevalent for the minperf algorithm when using the "PF B" method. Overall, the delay per user distributions become more spread apart when using the "PF B" method. The following table compares the percentiles and standard deviations of the delay per user distributions when using the two average rate calculation methods.

TABLE LXIX:
PERCENTILES AND STANDARD DEVIATIONS OF AVERAGE PACKET DELAY PER USER PER DROP FOR SCHEDULING ALGORITHMS IN THE ITU VEHICULAR B CHANNEL WITH 16 USERS PER SECTOR USING THE "PF A" AND "PF B" METHODS TO CALCULATE THE AVERAGE RATE

Calc. Method	"PF A" / "PF B"				
Algorithm	PF	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75	M-LWDF 2
10th percentile	25.1 / 21.3	13.4 / 12.8	16.7 / 15.0	20.5 / 18.1	30.5 / 27.7
25th	27.6 / 25.4	31.0 / 30.7	25.5 / 24.4	25.4 / 23.6	32.7 / 31.2
50th	35.5 / 37.8	76.6 / 95.1	46.2 / 54.6	37.5 / 41.9	38.8 / 39.2
75th	47.5 / 51.3	126.0 / 163.2	71.5 / 85.0	54.0 / 61.0	49.9 / 51.1
90th	82.9 / 94.0	235.8 / 330.5	136.6 / 173.3	99.5 / 118.4	76.5 / 80.3
Std. Dev. (kb/s)	57.1 / 65.5	160.2 / 255.2	94.8 / 129.2	69.4 / 86.5	53.7 / 56.8
Algorithm	minperf 5 kb/s	minperf 10 kb/s	minperf 15 kb/s	minperf 20 kb/s	minperf 25 kb/s
10th percentile	21.0 / 14.0	34.3 / 20.2	29.1 / 33.0	26.3 / 35.5	24.8 / 32.5
25th	52.3 / 60.4	39.0 / 55.8	33.1 / 45.5	29.8 / 39.9	27.7 / 35.9
50th	63.4 / 107.4	45.8 / 67.1	38.0 / 52.0	33.9 / 45.5	31.2 / 40.4
75th	76.7 / 129.2	53.2 / 78.3	44.9 / 59.9	39.1 / 51.4	36.3 / 45.5
90th	90.7 / 165.9	63.0 / 93.4	59.1 / 68.1	52.2 / 57.1	48.3 / 52.8
Std. Dev. (kb/s)	52.4 / 54.9	36.1 / 49.1	55.9 / 52.2	48.7 / 52.4	92.7 / 53.4
Algorithm	EXP2A	EXP2B	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75
10th percentile	33.4 / 31.7	25.5 / 21.7	13.9 / 13.3	17.1 / 15.5	20.7 / 18.5
25th	36.3 / 35.6	28.1 / 25.7	31.3 / 31.0	25.8 / 24.8	25.9 / 24.0
50th	40.7 / 40.9	35.8 / 38.0	72.9 / 89.5	45.9 / 53.8	37.9 / 41.9
75th	49.6 / 50.6	48.0 / 51.6	117.5 / 147.2	69.7 / 82.5	54.4 / 61.2
90th	74.9 / 76.7	82.9 / 93.9	216.5 / 289.3	133.0 / 162.6	99.2 / 116.1
Std. Dev. (kb/s)	50.2 / 51.9	55.1 / 64.4	147.8 / 215.4	95.8 / 120.1	67.5 / 83.2

From the table is can be seen that some of the minperf cases break the general trend of increasing standard deviations for the "PF B" method. Those exceptions are again due to users being ignored less often, and consequently experiencing lower delays. Hence, the tail at the upper end of the distribution is shorter, and so the standard deviations are smaller.

Figure F.10 in Appendix F compares the distributions of allocated slots per user per drop using the two calculation methods in the vehicular channel with 16 users per sector. There is overall a wider distribution of slots seen for all the algorithms when using the "PF B" method. There are two exceptions, however. The M-LWDF 2 and EXP2A algorithms, both of which place little emphasis on channel conditions, in fact show a tighter slot distribution when using the "PF B" method. Because of the low emphasis, users in very good channel conditions are not selected extremely often as they are with the other algorithms, so the distributions do not spread out for that reason. However, the more accurate calculation for the average rate does make it a bit more likely for a user to experience an early termination. Consequently, any given user does not tie up the channel for as long for a given packet, and so the slots can be allocated more fairly amongst the users, leading to the tighter distribution. The following table numerically compares the distributions.

TABLE LXX:
PERCENTILES AND STANDARD DEVIATIONS OF SLOTS ALLOCATED PER USER PER DROP FOR SCHEDULING ALGORITHMS IN THE ITU VEHICULAR B CHANNEL WITH 16 USERS PER SECTOR USING THE "PF A" AND "PF B" METHODS TO CALCULATE THE AVERAGE RATE

Calc. Method	"PF A" / "PF B"				
Algorithm	PF	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75	M-LWDF 2
10th percentile	961 / 803	299 / 197	536 / 396	765 / 602	960 / 1011
25th	1086 / 968	346 / 252	649 / 509	912 / 769	1022 / 1073
50th	1165 / 1102	452 / 360	829 / 682	1068 / 957	1120 / 1138
75th	1234 / 1314	974 / 977	1246 / 1292	1292 / 1376	1250 / 1217
90th	1294 / 1534	2979 / 3189	2168 / 2450	1610 / 1907	1373 / 1310
Std. Dev. (kb/s)	203.1 / 308.3	1633.8 / 1805.0	872.4 / 1095.0	385.0 / 570.4	217.3 / 197.5
Algorithm	minperf 5 kb/s	minperf 10 kb/s	minperf 15 kb/s	minperf 20 kb/s	minperf 25 kb/s
10th percentile	373 / 207	451 / 352	500 / 468	568 / 530	595 / 565
25th	502 / 312	634 / 505	705 / 603	778 / 648	814 / 705
50th	839 / 532	913 / 800	995 / 928	1086 / 918	1140 / 955
75th	1496 / 1081	1466 / 1463	1430 / 1481	1369 / 1438	1387 / 1431
90th	2091 / 2754	2173 / 2101	1939 / 2127	1795 / 2076	1678 / 1981
Std. Dev. (kb/s)	925.6 / 1671.6	679.4 / 989.2	579.8 / 675.6	490.5 / 626.0	453.9 / 573.8
Algorithm	EXP2A	EXP2B	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75
10th percentile	831 / 866	989 / 840	342 / 243	575 / 441	799 / 652
25th	937 / 965	1097 / 986	389 / 298	685 / 550	939 / 800
50th	1102 / 1101	1165 / 1106	487 / 401	845 / 706	1070 / 964
75th	1315 / 1288	1222 / 1296	976 / 978	1237 / 1288	1278 / 1369
90th	1502 / 1466	1277 / 1502	2869 / 3086	2103 / 2382	1575 / 1857
Std. Dev. (kb/s)	287.6 / 267.2	195.4 / 289.4	1558.7 / 1720.0	824.0 / 1031.1	361.9 / 536.8

To summarize, because of the numerous early terminations in the vehicular channel, the "PF B" method provides a significant improvement in throughput for those algorithms which calculate the average rate. Corresponding to the increased throughput are a decreased average delay per packet, an increased average delay per user, and wider distributions of throughputs, delays, and slots per user. As such, the fairness of the algorithms becomes somewhat lower, but the benefit of the more accurate calculation of the average rate and the resulting throughput probably makes the other drawbacks worth the tradeoff.

5.6. ITU Vehicular B Channel, Perfect One-slot Prediction and Margins

5.6.1. Effects of Perfect One-slot Prediction

As seen in the previous section, attempting to determine supportable formats based on an averaged SIR value causes a severe degradation of all aspects of system performance. Clearly, some kind of accounting for the Rayleigh fading is required in order to maintain a decent level of performance. To that end, the effect of perfect one-slot prediction shall be analyzed. As discussed both in the previous chapter and at the beginning of the current chapter, it is only feasible to predict the SIR value for the first transmission slot in the vehicular channel due to the fast fading rate. Margins are therefore still required for formats that take more than one slot to transmit a packet, as it is always possible for the SIR level to decrease in later slots and thus cause a packet error. The 1% packet error rate is still desired. For the formats that only use one slot, a negative margin is actually required. As seen with the perfect multislot prediction for the pedestrian channels, perfect knowledge that the SIR for a transmission will be always be at or above the threshold for a 1% PER for a format leads to packet error rates well below 1%. Thus, the negative margin is used to in effect overestimate the SIR level by a small amount. This makes packet errors somewhat more likely, as certain transmissions occur below the aforementioned threshold. Therefore, on average, the PER for the one-slot formats increases back to 1%.

On occasion with low numbers of users per sector, the one-slot prediction method produced rather high packet error rates for certain users who were on the cusp of being in outage. The first transmission slot SIR value including Rayleigh fading doesn't really give an indication about what the SIR value will be towards the end of a 16-slot packet transmission. Therefore, there were a few rare instances where the signal level would spike and a packet would be scheduled for a user, only to have that user return to outage and cause the packet to be in error. That user would therefore see a very high packet error rate, which would therefore skew the average packet error rate seen across the set of users. However, the average packet error rate for each format still remained around 1%, and when that user was filtered out, the average packet error rate per user also returned to about 1%. In addition, the falsely high PER per user was only seen for low numbers of users per sector; for larger numbers of users per sector, it was far more likely that there was a better user to choose from, so the near-outage user could be ignored until the SIR seen by that user improved sufficiently to allow successfully received packets. Therefore, despite the PER being artificially skewed to the high side on some occasions, the data was still deemed to be valid for analysis.

Figure G.1 in Appendix G shows the average sector throughput per drop as a function of the number of users per sector in the ITU vehicular B channel when using perfect one-slot prediction. The most obvious observation is that there is a dramatic improvement in throughput over the case of using outdated SIR_p values. The general shapes of the curves are similar to what was seen in the ITU pedestrian B channel. Interestingly, the throughput and multiuser gain achieved are even larger than that in the ITU pedestrian B channel. Using the example of the maxD algorithm, the throughput and multiuser gain at 32 users per sector are about 2260 kb/s and 3.9 respectively in the ITU vehicular B channel with perfect one-slot prediction, whereas they where about 1800 kb/s and 2.8 in the ITU pedestrian B channel with perfect multislot prediction.

The relative throughput of the M-LWDF 1 & 2, EXP1A, and EXP2A algorithms are improved compared to the other algorithms, while the slot-fair and packet fair algorithms are relatively somewhat lower. For example, notice that the slot-fair throughput crosses the PF throughput closer to 32 users per sector, compared to around 16 users per sector in the pedestrian channels. The cause of the relative position changes can likely be traced to the use of formats that take more slots to transmit at a given rate. When equalizing the number of slots each user receives, those users with intermediate signal levels would receive twice as much throughput in the pedestrian channels as in the vehicular channel for the same number of slots. Similarly for the packet-fair algorithm, when delivering a packet to a user, the intermediate users control the channel for twice as much time in the vehicular channel, meaning less service can be given to other users. (Note, however, that with the better prediction, the throughput is once again higher than that of the RR algorithm.) In comparison, the added delay seen by those intermediate users means that their packets would be chosen more often in the vehicular channel by the M-LWDF, EXP1A, and EXP2A algorithms, which emphasize delay. In the pedestrian channels, users who request lower rates would be chosen more often. Therefore, the throughput for the latter algorithms would increase relative to the pedestrian channels.

For the minperf algorithm, the better channel prediction increases the effective capacity of the channel. Therefore, users can receive larger than the minimum specified throughput more often, and as such the throughput does not drop off nearly as fast as when outdated SIR_p values were used for format determination. The shapes of the curves and the number of users at which the throughput increases/decreases are very similar to what was seen in the ITU pedestrian B channel. The throughputs maximize at higher values and minimize at lower values in the ITU vehicular B channel than in the ITU pedestrian B channel, however.

Figure G.2 in Appendix G shows the distributions of average throughput per user per drop for the vehicular channel with perfect one-slot prediction and 16 users per sector. Obviously, the throughputs per user improve along with the throughputs per sector, compared to the use of outdated SIR_p values. The range of throughputs seen is somewhere between the ranges seen in the ITU pedestrian A and B channels (i.e. smaller than the A channel, but larger than the B channel). Reflecting the sector throughputs, the users under the M-LWDF, EXP1A, and EXP2A algorithms see somewhat higher user throughputs relative to the other algorithms when compared to the pedestrian channels, while the slot-fair and packet-fair algorithms show relatively somewhat smaller user throughputs. One of the most noteworthy observations is that the probability of users not receiving any data at all is dramatically reduced compared with the case of using outdated SIR_p values. The minperf algorithm no longer shows any users being starved for service, and the probability of no service under the maxD algorithm drops to 4.88%. The probability of a user not receiving the minimum specified throughput under the minperf algorithm also drops to 2.06%, 2.88%, 4.06%, 4.88%, and 6.62% for the five analyzed minimums in increasing order. With the better prediction method, there is a much larger proportion of users receiving well above the minimum specified throughput, compared to the use of outdated SIR_p values.

The following table compares the statistics of the throughput distributions in the vehicular channel with perfect one-slot prediction.

TABLE LXXI:
MEAN, MEDIAN, AND STANDARD DEVIATION OF AVERAGE THROUGHPUT PER USER PER DROP DISTRIBUTIONS FOR SCHEDULING ALGORITHMS IN THE ITU VEHICULAR B CHANNEL WITH 16 USERS PER SECTOR AND PERFECT ONE-SLOT PREDICTION

Algorithm	RR	maxD	PF	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75
Mean (kb/s)	35.8	122.7	94.1	115.4	107.7	100.4
Median (kb/s)	35.4	45.0	89.6	73.9	84.9	88.6
Std. Dev. (kb/s)	19.3	168.5	56.6	113.2	87.8	69.3
Algorithm	M-LWDF 1	M-LWDF 2	slot-fair	packet-fair	minperf 5 kb/s	minperf 10 kb/s
Mean (kb/s)	80.4	68.6	88.3	66.8	103.6	79.5
Median (kb/s)	70.4	68.7	87.0	71.4	37.3	29.9
Std. Dev. (kb/s)	59.7	37.4	52.4	28.7	135.5	96.6
Algorithm	minperf 15 kb/s	minperf 20 kb/s	minperf 25 kb/s	EXP1A	EXP2A	EXP1B
Mean (kb/s)	58.2	45.2	40.7	61.5	54.1	122.2
Median (kb/s)	31.5	33.3	35.7	55.0	53.5	52.7
Std. Dev. (kb/s)	62.8	38.3	24.2	43.3	28.5	157.3

Algorithm	EXP2B	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75
Mean (kb/s)	93.5	114.9	107.7	99.7
Median (kb/s)	89.7	75.1	85.7	88.4
Std. Dev. (kb/s)	55.9	111.8	87.5	68.2

Figures G.3 and G.4 in Appendix G show the distributions of effective bit rates and the distributions of formats and number of slots used respectively for packets transmitted in the ITU vehicular B channel with perfect one-slot prediction and 16 users per sector. These two figures are the most beneficial in explaining the large throughput increase. With the much improved prediction method, the three highest rate formats can once again be supported. Note that most of the packets transmitted are sent with those formats and hence have high effective bit rates. Therefore, the throughput will naturally increase over the case when outdated SIR_p values are used. Also note that a larger proportion of packets are transmitted at those high rates in the vehicular channel than there were in the ITU pedestrian B channel. Thus, it makes sense for the resulting throughputs in the vehicular channel to be larger. The fast fading nature and high Doppler rate of the vehicular channel means that the range of variations seen on the user SIR values is larger than in the pedestrian channel. Consequently, an extremely high SIR value is obtained more often in the ITU vehicular B channel than in the ITU pedestrian B channel. This is confirmed by the distributions of the SIR values as shown in Chapter 4. Because of the fast fading, the signals do not remain at those high values for very long. However, if the fast fading can be accurately tracked, as it is with perfect one-slot prediction, those instantaneous high SIR values can be taken advantage of to transmit packets at very high data rates and consequently increase the system throughput relative to the ITU pedestrian B channel. Note that the signal levels and throughputs in the ITU pedestrian A channel are still higher than the other two cases, not particularly because of any fading issues, but simply because a larger percentage of the total transmitted power is captured at the receiver.

Figure G.5 in Appendix G shows the average delay per packet per drop as a function of the number of users per sector for the vehicular channel with perfect one-slot prediction. In general, the delays seen are smaller when compared to the use of outdated SIR_p values. The exception is the minperf algorithm at large numbers of users per sector. In that case, since users in poorer channel conditions are ignored less often thanks to the better prediction, resources are shared somewhat more fairly. This results in an overall increase in the delay an average packet sees. The same types of relative shifts that were seen in the throughputs are also seen in the delays. That is, compared to the pedestrian channels, the M-LWDF, EXP1A, and EXP2A packet delays are shorter and the slot-fair and packet-fair delays are longer relative to the other algorithms.

The relative shift is not quite as apparent for the average packet delay per user per drop curves shown in Figure G.6 in Appendix G. There is a noticeable drop in the relative user delay for the EXP1A and EXP2A algorithms compared to the pedestrian channels. In comparison, the slot-fair and packet-fair algorithms don't really change relative to the pedestrian channel results. There does appear to be an increase in delay for those two algorithms at 32 users per sector, but that is primarily a side effect of the algorithms dealing with a few near-outage users more than anything else. In any event, all the user delays are clearly lower than in the case of outdated SIR_p values. The largest reductions are in the maxD and the minperf algorithms, due to the better sharing of the system resources and the significantly reduced chance of a user not receiving any service.

Figures G.7 and G.8 in Appendix G show the distributions of packet delays for the vehicular channel with perfect one-slot prediction and 16 users per sector. Clearly, the delays seen are much lower than in the case of outdated SIR_p values. Compared to the ITU pedestrian B channel, the delays seem to be more varied. There appears to be a better chance of packets with very small delays, but also a much higher probability of significantly long delays in the vehicular channel. This is a reflection both on the larger throughput of the vehicular channel, and on the larger number of slots used to transmit packets at intermediate bit rates. The following table compares the percentiles and standard deviations of the packet delay distributions with perfect one-slot prediction.

TABLE LXXII:
PERCENTILES AND STANDARD DEVIATIONS OF PACKET DELAYS FOR SCHEDULING ALGORITHMS IN THE ITU VEHICULAR B CHANNEL WITH 16 USERS PER SECTOR AND PERFECT ONE-SLOT PREDICTION

Algorithm	RR	maxD	PF	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75
50th percentile	34.6	6.5	12.2	8.1	9.4	11.0
90th	49.9	25.7	49.1	37.9	42.6	46.2
95th	69.0	43.8	72.9	64.6	68.1	70.8
99th	159.5	164.5	182.3	213.1	198.1	189.6
99.5th	241.3	308.3	283.8	357.8	319.6	299.6
99.9th	726.7	1474.4	779.5	1050.9	903.0	830.0
99.95th	1159.3	2614.2	1146.3	1504.8	1323.3	1198.0
99.99th	2939.7	6878.2	2513.1	2886.1	2644.5	2513.8
Std. Dev.	72.28	149.02	74.40	86.14	80.12	76.12
Algorithm	M-LWDF 1	M-LWDF 2	slot-fair	packet-fair	minperf 5 kb/s	minperf 10 kb/s
50th percentile	19.5	24.1	12.3	16.2	7.6	9.6

90th	42.8	44.8	54.1	67.4	34.9	47.4
95th	60.3	57.9	82.8	100.2	62.9	88.5
99th	143.3	113.9	210.0	232.3	325.5	384.9
99.5th	206.4	163.3	311.5	337.1	602.6	580.0
99.9th	592.8	481.5	752.5	781.5	1361.4	1162.3
99.95th	986.6	850.6	1053.4	1071.7	1725.9	1455.3
99.99th	2718.9	2524.5	2256.7	2310.6	2727.2	2481.4
Std. Dev.	71.34	64.16	70.28	73.41	98.85	94.70

Algorithm	minperf 15 kb/s	minperf 20 kb/s	minperf 25 kb/s	EXP1A	EXP2A	EXP1B
50th percentile	12.0	13.5	13.6	28.1	30.6	7.7
90th	60.8	68.6	67.9	47.3	46.8	28.1
95th	120.1	141.0	139.6	59.7	57.1	48.4
99th	422.6	477.3	516.8	111.7	106.5	240.7
99.5th	606.7	686.6	748.2	155.7	155.0	460.6
99.9th	1194.0	1306.2	1403.9	472.9	480.0	1389.3
99.95th	1482.8	1640.3	1746.0	800.3	831.0	1997.3
99.99th	2532.8	2797.9	2829.7	2434.2	2503.9	3943.3
Std. Dev.	101.67	112.19	118.18	60.57	60.66	107.59

Algorithm	EXP2B	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75
50th percentile	12.9	8.6	9.9	11.6
90th	50.0	40.0	44.0	47.5
95th	73.7	69.1	70.6	72.3
99th	176.9	213.0	194.1	183.2
99.5th	263.5	329.1	296.0	275.3
99.9th	645.3	834.5	734.9	679.4
99.95th	945.4	1146.0	1031.4	977.9
99.99th	2399.3	2421.4	2402.3	2377.1
Std. Dev.	68.94	76.92	73.97	71.35

Comparing the standard deviations with those from the ITU pedestrian B channel, it is clear that the vehicular channel distributions show either a wider or a comparable variation in packet delay values.

Figure G.9 in Appendix G shows the distributions of average packet delay per user per drop for the vehicular channel with perfect one-slot prediction and 16 users per sector. Compared to the case of outdated SIR_p values, there is a general reduction in the delay per user seen. The exception appears to be the larger minimum specified throughputs for the minperf algorithm. Recall that under the earlier case there was a significant probability that a user would not receive any service for an entire drop. Those users were filtered out of the earlier distributions. For the perfect one-slot prediction case, those same users do receive service. As a result, the system resources are spread over a larger group of users in a fairer manner, resulting in longer delays for all the users. The delays and spread of distributions are still much larger than for the pedestrian channels. The following table compares the percentiles and standard deviations of the delay per user distributions with perfect one-slot prediction.

TABLE LXXIII:
PERCENTILES AND STANDARD DEVIATIONS OF AVERAGE PACKET DELAY PER USER FOR SCHEDULING ALGORITHMS IN THE ITU VEHICULAR B CHANNEL WITH 16 USERS PER SECTOR AND PERFECT ONE-SLOT PREDICTION

Algorithm	RR	maxD	PF	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75
10th percentile	32.8	9.2	16.0	11.1	12.6	14.3
25th	34.3	15.3	18.5	15.8	16.6	17.5

50th	37.7	42.1	26.0	30.9	27.3	26.5
75th	49.5	175.4	37.8	65.9	49.5	42.0
90th	89.0	782.9	60.5	136.8	90.4	70.9
Std. Dev.	74.9	1348.4	68.8	152.7	97.8	81.1
Algorithm	M-LWDF 1	M-LWDF 2	slot-fair	packet-fair	minperf 5 kb/s	minperf 10 kb/s
10th percentile	16.3	21.3	17.6	25.7	10.4	13.1
25th	20.5	24.4	19.0	27.5	18.5	24.9
50th	30.0	29.8	26.4	30.8	54.6	45.9
75th	51.8	38.5	38.6	36.1	89.3	65.4
90th	120.3	61.2	64.4	44.0	125.3	87.5
Std. Dev.	177.5	60.7	113.6	68.1	54.3	41.1
Algorithm	minperf 15 kb/s	minperf 20 kb/s	minperf 25 kb/s	EXP1A	EXP2A	EXP1B
10th percentile	18.9	27.4	27.5	21.4	26.4	9.6
25th	32.7	33.1	33.2	26.4	29.3	15.8
50th	43.3	43.5	44.6	33.4	33.5	42.4
75th	58.9	59.3	59.1	45.8	40.7	184.8
90th	74.3	73.3	69.1	76.8	62.3	831.6
Std. Dev.	43.1	87.9	52.0	59.8	48.0	914.5

Algorithm	EXP2B	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75
10th percentile	16.1	11.2	12.7	14.4
25th	18.7	15.8	16.6	17.6
50th	26.1	30.5	27.0	26.4
75th	37.7	64.2	49.2	41.7
90th	60.1	136.8	95.4	68.9
Std. Dev.	68.4	156.6	124.3	88.9

Figure G.10 in Appendix G shows the distributions of allocated slots per user per drop in the vehicular channel with perfect one-slot prediction and 16 users per sector. The tightness of the distributions compared to the case of outdated SIR_p values seems to vary a fair deal. Those algorithms that place a large emphasis on requested rates (e.g. maxD, maxD/PF 0.25) seem to show a tighter distribution with perfect one-slot prediction. The same is true for algorithms that place very little emphasis on the requested rate (e.g. EXP1A, EXP2A). However, those algorithms that place an intermediate emphasis on requested rates (e.g. PF, slot-fair) display a widening of their distributions under perfect one-slot prediction. The minperf algorithm also shows a wider distribution for its five cases. For the maxD algorithm, the tighter distribution stems from users receiving service instead of being starved. The intermediate algorithms and the minperf algorithm show wider distributions due to their increased throughput by allocating more service to users in better channel conditions. For the algorithms which place little emphasis on channel conditions, the better prediction leads to a higher probability of earlier terminations, such that users control the channel for smaller amounts of time, leading to a fairer allocation of slots. In any event, all the algorithms show a wider distribution when compared to the pedestrian channels. The following table compares the percentiles and standard deviations of the slot allocations with perfect one-slot prediction.

TABLE LXXIV:
PERCENTILES AND STANDARD DEVIATIONS OF ALLOCATED SLOTS PER USER PER DROP FOR SCHEDULING ALGORITHMS IN THE ITU VEHICULAR B CHANNEL WITH 16 USERS PER SECTOR AND PERFECT ONE-SLOT PREDICTION

Algorithm	RR	maxD	PF	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75

10th percentile	915	7	674	247	401	534
25th	1052	81	957	455	671	830
50th	1167	470	1189	834	1008	1118
75th	1262	1651	1340	1644	1522	1418
90th	1328	3283	1482	2415	1997	1692
Std. Dev.	217.2	1469.0	321.8	888.0	627.6	442.3
Algorithm	M-LWDF 1	M-LWDF 2	slot-fair	packet-fair	minperf 5 kb/s	minperf 10 kb/s
10th percentile	316	877	1022	732	235	344
25th	716	1076	1078	803	384	499
50th	1191	1183	1142	970	724	864
75th	1544	1254	1209	1325	1401	1629
90th	1792	1318	1258	1848	2701	2223
Std. Dev.	552.0	228.0	152.7	439.2	1079.5	785.7
Algorithm	minperf 15 kb/s	minperf 20 kb/s	minperf 25 kb/s	EXP1A	EXP2A	EXP1B
10th percentile	381	366	361	850	976	35
25th	509	486	477	1051	1074	139
50th	855	794	765	1135	1158	570
75th	1569	1506	1515	1212	1216	1689
90th	2369	2530	2564	1393	1269	3123
Std. Dev.	773.8	854.6	886.6	261.4	173.4	1343.3

Algorithm	EXP2B	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75
10th percentile	688	250	378	554
25th	962	474	674	846
50th	1189	847	1020	1123
75th	1335	1631	1519	1411
90th	1468	2396	2000	1677
Std. Dev.	313.5	872.7	629.3	430.4

To conclude, the use of perfect one-slot prediction in the vehicular channel causes a dramatic improvement in the throughput over the use of outdated SIR_p values. The sector throughput was even larger than that of the ITU pedestrian B channel, thanks to a more opportunistic use of the larger and faster variations in the vehicular channel. The throughputs per user were also improved, and the probability of receiving no service was greatly reduced. Under perfect one-slot prediction, the three highest rate formats could once again be supported, leading to the improved bit rates. The delays per packet and delays per user were also reduced compared with the use of outdated SIR_p values, although they were still larger than for the pedestrian channels.

Some of the algorithms shifted their results relative to the other algorithms when compared to the pedestrian channels. Namely, the EXP1A, EXP2A, and M-LWDF algorithms showed an increased position for their sector throughputs and lowered values for their average delays per packet relative to the other algorithms. Similarly, the slot-fair and packet-fair algorithms displayed relatively lower throughput and higher delays. The primary cause of this was due to the use of formats that took more slots to transmit the same data rate in the vehicular channel compared to the pedestrian channels. The larger number of slots used for users in intermediate channel conditions hindered the performance of the slot-fair and packet-fair algorithms, but the added delay that resulted for those users ended up improving the performance of the EXP1A, EXP2A, and M-LWDF algorithms, all of which place a higher emphasis on delays.

5.6.2. "PF A" vs. "PF B" Average Rate Calculation Performance

The figures in Appendix H show a comparison of the results obtained when using the "PF A" and "PF B" methods to calculate the average rate under perfect one-slot prediction. For the proportionally fair algorithm, once again there was very little difference to be seen in most of the results between the two methods. Although there was a visible increase in the average delay per user and a widening of the slot allocation distribution, these changes were not deemed significant enough to warrant simulating any of the other algorithms except for the minperf algorithm. The changes for the other algorithms would likely also have been mostly minor.

For the minperf algorithm, the resulting changes when using the "PF B" method were much the same as previously seen. That is, the sector throughput increased, the throughput per user distributions widened, with sharper increases around the minimum specified throughputs, the probability of not obtaining the minimum specified throughput increased, and a larger proportion of packets were sent using the highest rate formats. The average delay per packet and packet delay distributions generally decreased, with the distributions becoming narrower, the average delay per user generally increased, with the average packet delay per user distributions becoming wider, and the slot allocation distributions got wider as well. The reasons for these changes have been discussed in previous sections, and need not be repeated here.

There are a couple of exceptions to the above patterns, however. Note that the throughputs for the 20 and 25 kb/s cases of the "PF A" method in Figure H.1 of Appendix H are around the minimum of those two respective curves. Consequently, in those two cases, users in poorer channel conditions would start to be ignored to satisfy the minimum throughput for better users. As a result, the average delay per user and the distributions of delay per user end up taking on larger values under the "PF A" method. This also results in a widening of the delay per user and slot allocation distributions in those cases. Those same users are not ignored near as often under the "PF B" method, and so the values of average delay per user are less. Hence, the distributions of average packet delay per user and of allocated slots are somewhat tighter under the "PF B" method. This same effect was seen for the slot allocations at 25 kb/s in the ITU pedestrian B channel, although to a lesser extent.

The following tables compare the statistics and percentiles of the various distributions for the two calculation methods in the vehicular channel using perfect one-slot prediction.

TABLE LXXV:
STATISTICS FOR AVERAGE THROUGHPUT PER USER PER DROP DISTRIBUTIONS AND PROBABILITY OF NOT ACHIEVING MINIMUM SPECIFIED THROUGHPUT FOR THE MINIMUM PERFORMANCE SCHEDULING ALGORITHM IN THE ITU VEHICULAR B CHANNEL WITH 16 USERS PER SECTOR USING THE "PF A" AND "PF B" METHODS TO CALCULATE THE AVERAGE RATE AND PERFECT ONE-SLOT PREDICTION

Calc. Method	"PF A"					"PF B"				
minperf	5	10	15	20	25	5	10	15	20	25

C_k (kb/s)										
Mean (kb/s)	103.6	79.5	58.2	45.2	40.7	113.0	100.7	87.2	74.0	61.0
Median (kb/s)	37.3	29.9	31.5	33.3	35.7	41.0	37.5	34.2	33.0	33.1
Std. Dev. (kb/s)	135.5	96.6	62.8	38.3	24.2	150.9	129.3	105.9	83.2	60.7
P[Throughput <C_k] (%)	2.06	2.88	4.06	4.88	6.62	3.00	4.06	4.94	6.19	7.94

TABLE LXXVI:
PERCENTILES AND STANDARD DEVIATIONS OF PACKET DELAYS FOR THE MINIMUM PERFORMANCE SCHEDULING ALGORITHM IN THE ITU VEHICULAR B CHANNEL WITH 16 USERS PER SECTOR USING THE "PF A" AND "PF B" METHODS TO CALCULATE THE AVERAGE RATE AND PERFECT ONE-SLOT PREDICTION

Calc. Method	"PF A"					"PF B"				
C_k (kb/s) Percentile	5	10	15	20	25	5	10	15	20	25
50th (median)	7.6	9.6	12.0	13.5	13.6	7.2	7.9	9.0	10.6	12.1
90th	34.9	47.4	60.8	68.6	67.9	31.1	37.7	45.7	54.7	63.5
95th	62.9	88.5	120.1	141.0	139.6	56.4	70.4	87.7	106.6	122.9
99th	325.5	384.9	422.6	477.3	516.8	304.8	339.3	341.8	343.5	367.4
99.5th	602.6	580.0	606.7	686.6	748.2	581.5	527.7	495.5	492.1	529.4
99.9th	1361.4	1162.3	1194.0	1306.2	1403.9	1372.4	1143.9	1037.0	1005.2	1075.6
99.95th	1725.9	1455.3	1482.8	1640.3	1746.0	1745.9	1461.6	1342.7	1277.8	1363.6
99.99th	2727.2	2481.4	2532.8	2797.9	2829.7	2725.9	2506.7	2368.0	2378.3	2377.6
Std. Dev.	98.85	94.70	101.67	112.19	118.18	99.15	90.43	87.34	87.41	93.80

TABLE LXXVII:
PERCENTILES AND STANDARD DEVIATION OF AVERAGE PACKET DELAY PER USER FOR THE MINIMUM PERFORMANCE SCHEDULING ALGORITHM IN THE ITU VEHICULAR B CHANNEL WITH 16 USERS PER SECTOR USING THE "PF A" AND "PF B" METHODS TO CALCULATE THE AVERAGE RATE AND PERFECT ONE-SLOT PREDICTION

Calc. Method	"PF A"					"PF B"				
C_k (kb/s) Percentile	5	10	15	20	25	5	10	15	20	25
10th	10.4	13.1	18.9	27.4	27.5	9.8	10.7	12.1	14.3	18.8
25th	18.5	24.9	32.7	33.1	33.2	17.3	19.2	22.5	27.0	33.8
50th (median)	54.6	45.9	43.3	43.5	44.6	53.9	54.8	49.5	44.5	41.6
75th	89.3	65.4	58.9	59.3	59.1	121.0	79.4	64.2	54.5	50.0
90th	125.3	87.5	74.3	73.3	69.1	154.0	98.9	76.9	65.0	59.5
Std. Dev.	54.3	41.1	43.1	87.9	52.0	73.2	48.0	47.3	49.6	54.4

TABLE LXXVIII:
PERCENTILES AND STANDARD DEVIATION OF SLOTS ALLOCATED PER USER PER DROP FOR THE MINIMUM PERFORMANCE SCHEDULING ALGORITHM IN THE ITU VEHICULAR B CHANNEL WITH 16 USERS PER SECTOR USING THE "PF A" AND "PF B" METHODS TO CALCULATE THE AVERAGE RATE AND PERFECT ONE-SLOT PREDICTION

Calc. Method	"PF A"					"PF B"				
C_k (kb/s) Percentile	5	10	15	20	25	5	10	15	20	25
10th	235	344	381	366	361	205	316	406	467	499
25th	384	499	509	486	477	303	446	541	607	625
50th (median)	724	864	855	794	765	555	756	854	908	907
75th	1401	1629	1569	1506	1515	1500	1341	1493	1568	1511
90th	2701	2223	2369	2530	2564	2971	2586	2227	2066	2127
Std. Dev.	1079.5	785.7	773.8	854.6	886.6	1249.3	1006.9	793.2	661.5	634.9

5.7. ITU Vehicular B Channel, Perfect Multislot Prediction

5.7.1. Effects of Perfect Multislot Prediction

Perfect multislot prediction in the vehicular channel is treated much in the same way as in the pedestrian channels. That is, formats for each user are determined based on the average SIR value over all future slots where a transmission could potentially occur. The number of slots for which the SIR is averaged is dependent on the format being tested. As in the pedestrian channels, margins are not used, and the low values of packet error rates that result are accepted as is. However, in contrast to the pedestrian channels, the same formats that were allowed for previous vehicular channel results are also used here. That is, when selecting between two formats that nominally use a different number of slots to transfer a packet with a given bit rate, the format that uses the larger number of slots is always selected for the vehicular channel, even with the perfect multislot prediction as analyzed in this section.

It should be noted that in the case of the vehicular channel, perfect multislot prediction really is not a feasible method, even theoretically. As discussed earlier, it is only practical to be able to perfectly predict the SIR in at most the first transmission slot (three slots in the future) in the vehicular channel. The correlation of SIR values becomes insignificant beyond about 5.5 slots in the future. Thus, the analysis of perfect multislot prediction essentially just considers the hypothetical question of how much of a performance difference could be seen if predicting all transmission slots were somehow possible.

Figure I.1 in Appendix I shows the average sector throughput per drop as a function of the number of users per sector for the vehicular channel with perfect multislot prediction. There is an interesting effect to be observed with the throughput compared to the case of perfect one-slot prediction. For all the algorithms, at low numbers of users per sector (i.e. up to about 4 to 8 users per sector), the resulting throughput is larger under perfect multislot prediction. However, for larger numbers of users per sector, the throughput is actually larger when using perfect one-slot prediction. This difference can be explained by carefully examining how the margins affect the user selection. With low numbers of users per sector, it is somewhat less likely for any one particular user to have an extremely good signal. Therefore, the requested rates can be expected to have intermediate values. When margins are used, the mobiles must in effect underestimate the rates they can support in order to ensure the 1% packet error rate. This is particularly true in the vehicular channel, since there is very low correlation in the SIR values after only a few slots. Therefore, the required margins are particularly large in the vehicular channel. By using perfect multislot prediction, it is possible for those users with low to intermediate signal levels to request and receive a larger rate. Hence, the throughput at low numbers of users per sector increases. Compare this to the situation with large numbers of users per sector. The likelihood of a user with a very good signal is much larger. As such, the highest rate formats would be selected more often. This is particularly true for algorithms such as maxD, which exclusively considers the requested rate when making scheduling decisions. Recall that in the case of one-slot prediction, negative margins were used to obtain the target packet error rate of 1% for the highest rate formats. Under multislot prediction, the packet error rate is smaller, but the number of packets transmitted under the highest rates is also smaller, since the range of SIR values where those rates can be requested is reduced. With fewer transmissions at the highest rates, the throughput naturally decreases. Combined with this are the facts that a larger proportion of users request lower rates as the number of users per sector increases, the number of users considered to be in outage decreases, and the probability of an early termination decreases. These issues were discussed in more detail for the pedestrian channels, and cause a further reduction in throughput for algorithms that emphasize channel conditions to a smaller degree. This can be seen in the fact that the throughput gain at low numbers of users is largest and the throughput drop at high numbers of users is smallest between one-slot and multislot prediction for those algorithms which emphasize channel conditions the most. Similarly, those algorithms which place less emphasis on channel conditions show smaller throughput gains and larger throughput losses.

Figure I.2 in Appendix I shows the distributions of average throughput per user per drop for the vehicular channel with perfect multislot prediction and 16 users per sector. Reflecting the lower sector throughputs seen for 16 users per sector, the user throughput distributions are also generally shifted to lower values. By virtue of this shift, the distributions are a bit narrower than under one-slot prediction. There was also a slightly increased probability (5.5% versus 4.9% under one-slot prediction) of receiving no service under the maxD algorithm. This same effect was seen in the ITU pedestrian B channel. The better prediction method allows some of the intermediate users request higher rates and therefore receive more service to the exclusion of some of the other users in poorer channel conditions.

For the minperf algorithm, the distributions rise more sharply under perfect multislot prediction. This is likely due to the lessened likelihood of early terminations that makes the weaker users operate closer to the minimum specified throughput. Fortunately, the extra channel capacity caused by the better prediction also makes it easier to meet that minimum for a given user. For example, any given user will appear to be in outage for a smaller proportion of the time. Hence, the probability of not meeting the specified minimum decreases. For the five analyzed minimums in increasing order, the probabilities of not obtaining the minimum throughput in the vehicular channel under perfect multislot prediction are 0.81%, 1.94%, 2.56%, 3.50%, and 5.63%, respectively.

One exception to the shift towards lower throughputs is the EXP1B algorithm. At the lower half of the distribution, the throughputs increase as those users thought to be in outage before are now discovered to be able to support a non-zero bit rate. Hence, the throughputs of those users increase. A similar shift can be seen for the other algorithms, but the shift only occurs at the very low end tail of the distributions and is not quite as significant as for the EXP1B algorithm.

The following table compares the statistics of the throughput distributions.

TABLE LXXIX:
MEAN, MEDIAN, AND STANDARD DEVIATION OF AVERAGE THROUGHPUT PER USER PER DROP DISTRIBUTIONS FOR SCHEDULING ALGORITHMS IN THE ITU VEHICULAR B CHANNEL WITH 16 USERS PER SECTOR AND PERFECT MULTISLOT PREDICTION

<table>
<tr><th>Value</th><th colspan="12">Perfect One-slot Prediction / Perfect Multislot Prediction</th></tr>
<tr><td>Algorithm</td><td colspan="2">RR</td><td colspan="2">maxD</td><td colspan="2">PF</td><td colspan="2">maxD/PF 0.25</td><td colspan="2">maxD/PF 0.5</td><td colspan="2">maxD/PF 0.75</td></tr>
<tr><td>Mean (kb/s)</td><td colspan="2">35.8 / 33.7</td><td colspan="2">122.7 / 121.0</td><td colspan="2">94.1 / 87.7</td><td colspan="2">115.4 / 112.0</td><td colspan="2">107.7 / 102.6</td><td colspan="2">100.4 / 94.4</td></tr>
<tr><td>Median (kb/s)</td><td colspan="2">35.4 / 35.1</td><td colspan="2">45.0 / 44.0</td><td colspan="2">89.6 / 87.2</td><td colspan="2">73.9 / 77.2</td><td colspan="2">84.9 / 86.4</td><td colspan="2">88.6 / 88.0</td></tr>
<tr><td>Std. Dev. (kb/s)</td><td colspan="2">19.3 / 14.6</td><td colspan="2">168.5 / 167.4</td><td colspan="2">56.6 / 46.3</td><td colspan="2">113.2 / 102.9</td><td colspan="2">87.8 / 75.4</td><td colspan="2">69.3 / 57.4</td></tr>
<tr><td>Algorithm</td><td colspan="2">M-LWDF 1</td><td colspan="2">M-LWDF 2</td><td colspan="2">slot-fair</td><td colspan="2">packet-fair</td><td colspan="2">minperf 5 kb/s</td><td colspan="2">minperf 10 kb/s</td></tr>
<tr><td>Mean (kb/s)</td><td colspan="2">80.4 / 74.7</td><td colspan="2">68.6 / 59.2</td><td colspan="2">88.3 / 88.0</td><td colspan="2">66.8 / 59.0</td><td colspan="2">103.6 / 96.2</td><td colspan="2">79.5 / 72.3</td></tr>
<tr><td>Median (kb/s)</td><td colspan="2">70.4 / 71.9</td><td colspan="2">68.7 / 62.4</td><td colspan="2">87.0 / 87.0</td><td colspan="2">71.4 / 61.6</td><td colspan="2">37.3 / 34.1</td><td colspan="2">29.9 / 27.5</td></tr>
<tr><td>Std. Dev. (kb/s)</td><td colspan="2">59.7 / 47.4</td><td colspan="2">37.4 / 27.2</td><td colspan="2">52.4 / 49.7</td><td colspan="2">28.7 / 22.6</td><td colspan="2">135.5 / 128.3</td><td colspan="2">96.6 / 92.4</td></tr>
<tr><td>Algorithm</td><td colspan="2">minperf 15 kb/s</td><td colspan="2">minperf 20 kb/s</td><td colspan="2">minperf 25 kb/s</td><td colspan="2">EXP1A</td><td colspan="2">EXP2A</td><td colspan="2">EXP1B</td></tr>
<tr><td>Mean (kb/s)</td><td colspan="2">58.2 / 54.0</td><td colspan="2">45.2 / 44.3</td><td colspan="2">40.7 / 41.3</td><td colspan="2">61.5 / 46.9</td><td colspan="2">54.1 / 42.6</td><td colspan="2">122.2 / 120.3</td></tr>
<tr><td>Median (kb/s)</td><td colspan="2">31.5 / 25.0</td><td colspan="2">33.3 / 28.2</td><td colspan="2">35.7 / 31.6</td><td colspan="2">55.0 / 46.7</td><td colspan="2">53.5 / 44.3</td><td colspan="2">52.7 / 57.6</td></tr>
<tr><td>Std. Dev. (kb/s)</td><td colspan="2">62.8 / 64.2</td><td colspan="2">38.3 / 44.4</td><td colspan="2">24.2 / 32.8</td><td colspan="2">43.3 / 25.0</td><td colspan="2">28.5 / 19.0</td><td colspan="2">157.3 / 150.1</td></tr>
<tr><td>Algorithm</td><td colspan="3">EXP2B</td><td colspan="3">maxD/PF/EXP 0.25</td><td colspan="3">maxD/PF/EXP 0.5</td><td colspan="3">maxD/PF/EXP 0.75</td></tr>
<tr><td>Mean (kb/s)</td><td colspan="3">93.5 / 87.1</td><td colspan="3">114.9 / 110.4</td><td colspan="3">107.7 / 101.1</td><td colspan="3">99.7 / 93.1</td></tr>
<tr><td>Median (kb/s)</td><td colspan="3">89.7 / 87.0</td><td colspan="3">75.1 / 77.4</td><td colspan="3">85.7 / 86.5</td><td colspan="3">88.4 / 87.3</td></tr>
<tr><td>Std. Dev. (kb/s)</td><td colspan="3">55.9 / 45.7</td><td colspan="3">111.8 / 99.9</td><td colspan="3">87.5 / 73.0</td><td colspan="3">68.2 / 55.9</td></tr>
</table>

Figures I.3 and I.4 in Appendix I show the distributions of effective bit rates and the distributions of formats and number of slots used respectively for packets transmitted in the ITU vehicular B channel with perfect multislot prediction and 16 users per sector. There is a noticeable drop in the number of packets received at the two highest bit rates, thanks to the removal of the negative margins for DRC10 and DRC12. There is also a much higher proportion of packets received at 1228.8 kb/s, which is the highest rate of the two-slot formats (DRC11). Due to perfect multislot prediction, that format can be supported much more often than under one-slot prediction, and consequently more packets are received with that rate and format. Most of the intermediate formats and rates see a decrease in their proportions, as packets are sent via higher rates. One exception is an increase at 409.6 kb/s. This is the case of a three-slot early termination of DRC6. Under one-slot prediction, the majority of transmissions using DRC6 ended after 2 slots, for an effective rate of 1228.8 kb/s. Under multislot prediction, seeing that the packet can be received at 1228.8 kb/s, the users directly request that rate instead of 614.4 kb/s. Thus, any transmissions that do occur under DRC6 instead terminate after 3 or 4 slots, leading to the increase of packets received at 409.6 kb/s. For a similar reason, there is an increase in the proportions of packets received at the lowest bit rates, since very early terminations of DRC1 to DRC3 also no longer occur.

Figure I.5 in Appendix I shows the average delay per packet per drop as a function of the number of users per sector for the vehicular channel with perfect multislot prediction. There is an increase in the delay seen under all the scheduling algorithms. The fact of the increase and the relative amount of increase in delay for each of the algorithms is similar to what was seen when switching from outdated channel information to perfect multislot prediction in the pedestrian channels. The reasons for the increases and why the delays for certain algorithms increase more than others are also essentially the same as in the pedestrian channels, so they need not be repeated here.

The changes in the average packet delay per user per drop vs. the number of users per sector curves seen in Figure I.6 in Appendix I for the most part follow the same kind of throughput-delay relationship seen several times before. For the algorithms which place at least a moderate emphasis on channel conditions, the increased sector throughput at low numbers of users per sector corresponds to an increase in the average delay per user seen. For larger numbers of users per sector, the decrease in throughput under perfect multislot prediction is reflected in a decreased packet delay per user. The notable exceptions to this general statement are the RR, maxD, M-LWDF 2, minperf, EXP1A, and EXP2A algorithms, all of which show an increase in the delay per user for all numbers of users per sector. In the case of the maxD algorithm, the additional delay is due to a larger probability of not receiving any service under perfect multislot prediction. For the other algorithms, due to the various fairness guarantees they provide, a reasonably large proportion (compared to the other algorithms) of packets are sent using the lowest rate formats. Most of these packets tend to go to the same user or subset of users. Under perfect multislot prediction, very early terminations of those transmissions no longer occur. Hence, the average delay experienced by those packets, and by extension the users receiving those packets, is larger under multislot prediction. Consequently, the average delay per user value for those algorithms is larger for all numbers of users.

Figures I.7 and I.8 in Appendix I show the distributions of packet delays for the vehicular channel with perfect multislot prediction and 16 users per sector. As with the average delay per packet curves, the distributions of packet delays also show larger values and wider distributions compared to the case of one-slot prediction. In both cases the reasons for the increases are the same, with a lack of early terminations playing a prominent role. However, the probability of extremely long packet delays (on the order of a couple thousand slots) is decreased under multislot prediction, as users do not appear to be in outage for as long a period of time, and consequently those packets are served sooner. The following table compares the percentiles and the standard deviations of the packet delay distributions in the vehicular channel with perfect multislot prediction.

TABLE LXXX:
PERCENTILES AND STANDARD DEVIATIONS OF PACKET DELAYS FOR SCHEDULING ALGORITHMS IN THE ITU VEHICULAR B CHANNEL WITH 16 USERS PER SECTOR AND PERFECT MULTISLOT PREDICTION

Value	Perfect One-slot Prediction / Perfect Multislot Prediction					
Algorithm	RR	maxD	PF	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75
50th percentile	34.6 / 49.2	6.5 / 6.8	12.2 / 13.9	8.1 / 8.6	9.4 / 10.6	11.0 / 12.3
90th	49.9 / 68.2	25.7 / 25.8	49.1 / 57.0	37.9 / 41.2	42.6 / 47.7	46.2 / 52.8
95th	69.0 / 78.1	43.8 / 43.4	72.9 / 84.6	64.6 / 69.4	68.1 / 75.5	70.8 / 80.3
99th	159.5 / 116.9	164.5 / 160.6	182.3 / 209.2	213.1 / 226.9	198.1 / 216.3	189.6 / 211.2
99.5th	241.3 / 175.7	308.3 / 303.3	283.8 / 319.2	357.8 / 383.3	319.6 / 349.7	299.6 / 331.9

99.9th	726.7 / 667.9	1474.4 / 1514.5	779.5 / 846.1	1050.9 / 1072.2	903.0 / 950.9	830.0 / 886.2
99.95th	1159.3 / 1181.9	2614.2 / 2691.7	1146.3 / 1183.8	1504.8 / 1485.5	1323.3 / 1313.5	1198.0 / 1235.4
99.99th	2939.7 / 3358.3	6878.2 / 7241.1	2513.1 / 2358.5	2886.1 / 2676.4	2644.5 / 2496.9	2513.8 / 2410.9
Std. Dev.	72.28 / 75.31	149.02 / 151.65	74.40 / 74.56	86.14 / 82.65	80.12 / 77.68	76.12 / 75.53
Algorithm	M-LWDF 1	M-LWDF 2	slot-fair	packet-fair	minperf 5 kb/s	minperf 10 kb/s
50th percentile	19.5 / 24.6	24.1 / 32.0	12.3 / 12.7	16.2 / 18.1	7.6 / 7.6	9.6 / 8.9
90th	42.8 / 48.6	44.8 / 57.0	54.1 / 58.7	67.4 / 80.5	34.9 / 36.7	47.4 / 49.1
95th	60.3 / 68.2	57.9 / 73.5	82.8 / 91.9	100.2 / 122.8	62.9 / 67.6	88.5 / 93.7
99th	143.3 / 159.9	113.9 / 128.9	210.0 / 244.6	232.3 / 340.1	325.5 / 441.4	384.9 / 544.9
99.5th	206.4 / 224.0	163.3 / 165.6	311.5 / 356.0	337.1 / 519.5	602.6 / 739.5	580.0 / 818.7
99.9th	592.8 / 520.8	481.5 / 421.6	752.5 / 798.8	781.5 / 1155.6	1361.4 / 1518.7	1162.3 / 1576.9
99.95th	986.6 / 800.1	850.6 / 775.1	1053.4 / 1079.0	1071.7 / 1531.6	1725.9 / 1883.3	1455.3 / 1974.8
99.99th	2718.9 / 2274.1	2524.5 / 2362.2	2256.7 / 2198.6	2310.6 / 2823.4	2727.2 / 2805.5	2481.4 / 3059.4
Std. Dev.	71.34 / 63.16	64.16 / 62.27	70.28 / 74.80	73.41 / 97.96	98.85 / 112.32	94.70 / 125.53
Algorithm	minperf 15 kb/s	minperf 20 kb/s	minperf 25 kb/s	EXP1A	EXP2A	EXP1B
50th percentile	12.0 / 11.5	13.5 / 13.6	13.6 / 14.4	28.1 / 40.5	30.6 / 43.3	7.7 / 8.1
90th	60.8 / 59.7	68.6 / 64.8	67.9 / 66.0	47.3 / 64.3	46.8 / 65.6	28.1 / 30.3
95th	120.1 / 129.2	141.0 / 149.4	139.6 / 152.8	59.7 / 77.8	57.1 / 77.9	48.4 / 50.3
99th	422.6 / 667.0	477.3 / 768.4	516.8 / 811.3	111.7 / 120.9	106.5 / 116.3	240.7 / 264.3
99.5th	606.7 / 983.7	686.6 / 1104.1	748.2 / 1146.9	155.7 / 148.9	155.0 / 146.7	460.6 / 468.0
99.9th	1194.0 / 1851.9	1306.2 / 1996.2	1403.9 / 1982.7	472.9 / 465.3	480.0 / 489.7	1389.3 / 1343.2
99.95th	1482.8 / 2232.7	1640.3 / 2399.8	1746.0 / 2380.8	800.3 / 874.9	831.0 / 896.8	1997.3 / 1998.1
99.99th	2532.8 / 3263.0	2797.9 / 3432.1	2829.7 / 3586.1	2434.2 / 2701.1	2503.9 / 2808.4	3943.3 / 3620.2
Std. Dev.	101.67 / 146.36	112.19 / 161.22	118.18 / 165.44	60.57 / 66.43	60.66 / 67.68	107.59 / 101.87

Algorithm	EXP2B	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75
50th percentile	12.9 / 15.0	8.6 / 9.3	9.9 / 11.6	11.6 / 13.5
90th	50.0 / 58.5	40.0 / 45.2	44.0 / 50.9	47.5 / 55.2
95th	73.7 / 85.1	69.1 / 76.5	70.6 / 79.6	72.3 / 82.5
99th	176.9 / 197.3	213.0 / 222.9	194.1 / 207.6	183.2 / 200.4
99.5th	263.5 / 284.6	329.1 / 336.6	296.0 / 307.3	275.3 / 294.8
99.9th	645.3 / 646.5	834.5 / 781.1	734.9 / 716.8	679.4 / 677.3
99.95th	945.4 / 891.7	1146.0 / 1035.0	1031.4 / 948.8	977.9 / 917.7
99.99th	2399.3 / 2140.5	2421.4 / 2107.6	2402.3 / 2060.5	2377.1 / 2124.8
Std. Dev.	68.94 / 67.40	76.92 / 70.17	73.97 / 67.93	71.35 / 67.55

Figure I.9 in Appendix I shows the distributions of average packet delay per user per drop for the vehicular channel with perfect one-slot prediction and 16 users per sector. There seems to be a general upwards shift in the distributions for the algorithms. Of particular note is the minperf algorithm, for which the two-step rise in the distribution is more apparent than with one-slot prediction. For the algorithms that display a decreased average packet delay per user per drop value in the earlier curves as a function of the number of users per sector, the probability of long delays at the upper end of the distributions is smaller under perfect multislot prediction. The length of the tail at the upper end of the distributions for those algorithms is also shorter. Because of this decreased likelihood of long delays per user, the average delay per user is reduced. For the other algorithms that do not show a decreased average delay per user, the probabilities at the upper end of the distributions do not show the same magnitude of reduction, although the length of the tail is still usually somewhat reduced. Consequently, in most cases, the distribution is somewhat tighter using perfect multislot prediction.

The following table compares the percentiles and standard deviations of the distributions of delay per user for perfect multislot prediction.

TABLE LXXXI:
PERCENTILES AND STANDARD DEVIATIONS OF AVERAGE PACKET DELAY PER USER FOR SCHEDULING ALGORITHMS IN THE ITU VEHICULAR B CHANNEL WITH 16 USERS PER SECTOR AND PERFECT MULTISLOT PREDICTION

Value	Perfect One-slot Prediction / Perfect Multislot Prediction					
Algorithm	RR	maxD	PF	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75
10th percentile	32.8 / 45.4	9.2 / 9.3	16.0 / 18.0	11.1 / 11.6	12.6 / 13.6	14.3 / 15.8
25th	34.3 / 47.6	15.3 / 15.8	18.5 / 21.1	15.8 / 16.7	16.6 / 18.2	17.5 / 19.6
50th	37.7 / 51.5	42.1 / 43.1	26.0 / 29.1	30.9 / 32.1	27.3 / 29.5	26.5 / 29.0
75th	49.5 / 59.2	175.4 / 178.2	37.8 / 40.8	65.9 / 64.4	49.5 / 49.2	42.0 / 43.4
90th	89.0 / 80.3	782.9 / 786.5	60.5 / 63.9	136.8 / 128.3	90.4 / 89.2	70.9 / 72.0
Std. Dev.	74.9 / 44.3	1348.4 / 1357.1	68.8 / 47.9	152.7 / 101.4	97.8 / 71.1	81.1 / 56.2
Algorithm	M-LWDF 1	M-LWDF 2	slot-fair	packet-fair	minperf 5 kb/s	minperf 10 kb/s
10th percentile	16.3 / 19.8	21.3 / 27.8	17.6 / 17.8	25.7 / 30.0	10.4 / 10.8	13.1 / 13.8
25th	20.5 / 24.2	24.4 / 31.5	19.0 / 20.0	27.5 / 33.3	18.5 / 20.2	24.9 / 27.8
50th	30.0 / 33.4	29.8 / 37.1	26.4 / 28.9	30.8 / 38.2	54.6 / 65.0	45.9 / 68.0
75th	51.8 / 53.9	38.5 / 46.9	38.6 / 43.1	36.1 / 45.7	89.3 / 127.2	65.4 / 89.2
90th	120.3 / 114.6	61.2 / 68.1	64.4 / 83.6	44.0 / 59.4	125.3 / 162.3	87.5 / 107.6
Std. Dev.	177.5 / 106.6	60.7 / 38.5	113.6 / 179.5	68.1 / 45.1	54.3 / 68.0	41.1 / 40.0
Algorithm	minperf 15 kb/s	minperf 20 kb/s	minperf 25 kb/s	EXP1A	EXP2A	EXP1B
10th percentile	18.9 / 19.7	27.4 / 28.3	27.5 / 37.2	21.4 / 33.5	26.4 / 37.9	9.6 / 9.8
25th	32.7 / 41.9	33.1 / 47.3	33.2 / 45.9	26.4 / 38.1	29.3 / 41.3	15.8 / 16.3
50th	43.3 / 60.9	43.5 / 57.4	44.6 / 54.9	33.4 / 44.4	33.5 / 46.2	42.4 / 42.3
75th	58.9 / 75.4	59.3 / 70.7	59.1 / 66.0	45.8 / 56.4	40.7 / 55.2	184.8 / 181.8
90th	74.3 / 91.3	73.3 / 82.2	69.1 / 75.3	76.8 / 81.4	62.3 / 74.4	831.6 / 767.8
Std. Dev.	43.1 / 32.2	87.9 / 30.6	52.0 / 34.4	59.8 / 45.5	48.0 / 37.9	914.5 / 793.6
Algorithm	EXP2B	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75		
10th percentile	16.1 / 18.2	11.2 / 11.8	12.7 / 13.9	14.4 / 16.1		
25th	18.7 / 21.3	15.8 / 17.0	16.6 / 18.4	17.6 / 19.8		
50th	26.1 / 29.2	30.5 / 32.0	27.0 / 29.7	26.4 / 29.2		
75th	37.7 / 40.8	64.2 / 62.5	49.2 / 49.0	41.7 / 43.4		
90th	60.1 / 64.0	136.8 / 122.2	95.4 / 86.9	68.9 / 71.9		
Std. Dev.	68.4 / 46.3	156.6 / 93.8	124.3 / 65.5	88.9 / 52.8		

Figure I.10 in Appendix I shows the distributions of allocated slots per user per drop in the vehicular channel with perfect multislot prediction and 16 users per sector. Most of the distributions seem to tighten under multislot prediction. The exceptions are largely the same algorithms that showed a consistent increase in the average packet delay per user seen earlier. Namely, the RR, maxD, minperf, packet-fair, EXP1A, and EXP2A algorithms displayed more spread out slot distributions. The lack of early terminations is again the key to the spread. Without the early terminations, users in poorer channel conditions require more slots to receive their packet correctly. The additional slots allocated to those users mean fewer slots allocated to other users, and hence a wider variation in the total number of slots allocated to each user. In the case of the maxD algorithm, the increased probability of no (or very little) service rather than the lack of early terminations is the main component behind the wider distribution.

The following table compares the percentiles and standard deviations of the slot distributions with perfect multislot prediction.

TABLE LXXXII:
PERCENTILES AND STANDARD DEVIATIONS OF ALLOCATED SLOTS PER USER PER DROP FOR SCHEDULING ALGORITHMS IN THE ITU VEHICULAR B CHANNEL WITH 16 USERS PER SECTOR AND PERFECT MULTISLOT PREDICTION

Value	Perfect One-slot Prediction / Perfect Multislot Prediction					
Algorithm	RR	maxD	PF	maxD/PF 0.25	maxD/PF 0.5	maxD/PF 0.75
10th percentile	915 / 707	7 / 5	674 / 983	247 / 362	401 / 606	534 / 811
25th	1052 / 822	81 / 71	957 / 1063	455 / 531	671 / 769	830 / 937
50th	1167 / 1029	470 / 462	1189 / 1150	834 / 895	1008 / 1055	1118 / 1127
75th	1262 / 1384	1651 / 1644	1340 / 1204	1644 / 1556	1522 / 1395	1418 / 1287
90th	1328 / 1693	3283 / 3290	1482 / 1249	2415 / 2211	1997 / 1765	1692 / 1449
Std. Dev.	217.2 / 388.9	1469.0 / 1489.0	321.8 / 138.3	888.0 / 749.6	627.6 / 455.5	442.3 / 257.0
Algorithm	M-LWDF 1	M-LWDF 2	slot-fair	packet-fair	minperf 5 kb/s	minperf 10 kb/s
10th percentile	316 / 562	877 / 928	1022 / 1038	732 / 568	235 / 228	344 / 329
25th	716 / 870	1076 / 1017	1078 / 1074	803 / 661	384 / 356	499 / 454
50th	1191 / 1208	1183 / 1099	1142 / 1129	970 / 844	724 / 719	864 / 786
75th	1544 / 1410	1254 / 1197	1209 / 1182	1325 / 1263	1401 / 1577	1629 / 1627
90th	1792 / 1544	1318 / 1376	1258 / 1223	1848 / 2200	2701 / 2596	2223 / 2410
Std. Dev.	552.0 / 373.6	228.0 / 185.5	152.7 / 106.1	439.2 / 711.9	1079.5 / 1041.6	785.7 / 858.5
Algorithm	minperf 15 kb/s	minperf 20 kb/s	minperf 25 kb/s	EXP1A	EXP2A	EXP1B
10th percentile	381 / 351	366 / 352	361 / 365	850 / 837	976 / 746	35 / 47
25th	509 / 467	486 / 459	477 / 467	1051 / 947	1074 / 851	139 / 160
50th	855 / 723	794 / 726	765 / 715	1135 / 1072	1158 / 1020	570 / 615
75th	1569 / 1520	1506 / 1463	1515 / 1401	1212 / 1246	1216 / 1307	1689 / 1706
90th	2369 / 2541	2530 / 2579	2564 / 2605	1393 / 1494	1269 / 1696	3123 / 2997
Std. Dev.	773.8 / 911.5	854.6 / 972.6	886.6 / 973.4	261.4 / 273.3	173.4 / 369.5	1343.3 / 1273.4
Algorithm	EXP2B	maxD/PF/EXP 0.25	maxD/PF/EXP 0.5	maxD/PF/EXP 0.75		
10th percentile	688 / 995	250 / 411	378 / 654	554 / 851		
25th	962 / 1073	474 / 564	674 / 793	846 / 955		
50th	1189 / 1147	847 / 907	1020 / 1059	1123 / 1125		
75th	1335 / 1196	1631 / 1524	1519 / 1372	1411 / 1269		
90th	1468 / 1238	2396 / 2166	2000 / 1720	1677 / 1425		
Std. Dev.	313.5 / 130.9	872.7 / 711.4	629.3 / 424.3	430.4 / 234.8		

To conclude, the elimination of the margins when switching to perfect multislot prediction caused a variety of effects in the vehicular channel. For low numbers of users per sector, the throughputs increased for all the algorithms. However, at larger numbers of users per sector, the throughput decreased when using multislot prediction. This latter effect could be traced to the fact that negative margins were used for the highest rate formats under one-slot prediction, and consequently those formats were requested more often under perfect one-slot prediction than under perfect multislot prediction. The removal of the negative margins also caused a reduction in the average throughputs per user and the proportion of packets received at the highest possible bit rates. In terms of delays, all the algorithms showed an increase in the average delay per packet, due in part to the fact that early terminations were much less likely. The average packet delay per user for the most part echoed the throughput curves; that is, an increased throughput when using multislot prediction corresponded to an increased delay per user, and vice versa. The exceptions were mostly those algorithms that showed a comparatively large proportion of packets transmitted to users with the lowest rate formats. The lack of early terminations resulted in larger delays for certain users, and consequently a larger average delay per user overall. Those same algorithms displayed a wider distribution of allocated slots per user, whereas the other algorithms showed narrower distributions of slots. The excess delay for those certain algorithms did not affect the distributions of delay per user, however, as almost all of the algorithms showed a smaller standard deviation for their distributions of delay per user under perfect multislot prediction.

5.7.2. "PF A" vs. "PF B" Average Rate Calculation Performance

As in the case of perfect multislot prediction in the pedestrian channels, running the simulations a second time for all the algorithms involving an average rate calculation required a negligible amount of extra time. The results of those simulations when using the "PF B" method to calculate the average rate are shown in Appendix J. As seen several times before, only the minperf algorithm displayed any significant changes in the results when using the "PF B" method. The ensuing changes were all similar to what was seen earlier for comparisons of the two calculation methods. Therefore, it is not required to go into any detail describing the reasons for those changes, as they are the same as previously discussed. Only the tables comparing the various percentiles and statistics for the distributions of the minperf algorithm shall be given here.

TABLE LXXXIII:
STATISTICS FOR AVERAGE THROUGHPUT PER USER PER DROP DISTRIBUTIONS AND PROBABILITY OF NOT ACHIEVING MINIMUM SPECIFIED THROUGHPUT FOR THE MINIMUM PERFORMANCE SCHEDULING ALGORITHM IN THE ITU VEHICULAR B CHANNEL WITH 16 USERS PER SECTOR USING THE "PF A" AND "PF B" METHODS TO CALCULATE THE AVERAGE RATE AND PERFECT MULTISLOT PREDICTION

Calc. Method	"PF A"					"PF B"				
minperf C_k (kb/s)	5	10	15	20	25	5	10	15	20	25
Mean (kb/s)	96.2	72.3	54.0	44.3	41.3	100.9	80.7	63.2	51.3	44.4
Median (kb/s)	34.1	27.5	25.0	28.2	31.6	35.9	30.1	26.9	27.6	31.1
Std. Dev. (kb/s)	128.3	92.4	64.2	44.4	32.8	135.2	103.7	76.8	55.7	39.6
P[Throughput $<C_k$] (%)	0.81	1.94	2.56	3.50	5.63	1.25	2.38	3.19	4.50	6.31

TABLE LXXXIV:
PERCENTILES AND STANDARD DEVIATIONS OF PACKET DELAYS FOR THE MINIMUM PERFORMANCE SCHEDULING ALGORITHM IN THE ITU VEHICULAR B CHANNEL WITH 16 USERS PER SECTOR USING THE "PF A" AND "PF B" METHODS TO CALCULATE THE AVERAGE RATE AND PERFECT MULTISLOT PREDICTION

Calc. Method	"PF A"					"PF B"				
C_k (kb/s) / Percentile	5	10	15	20	25	5	10	15	20	25
50th (median)	7.6	8.9	11.5	13.6	14.4	7.5	8.4	10.2	12.3	13.9
90th	36.7	49.1	59.7	64.8	66.0	35.2	46.6	57.4	64.0	67.7
95th	67.6	93.7	129.2	149.4	152.8	65.6	88.7	120.4	144.1	153.3
99th	441.4	544.9	667.0	768.4	811.3	417.4	484.7	557.7	657.1	750.3
99.5th	739.5	818.7	983.7	1104.1	1146.9	698.7	723.0	835.9	963.2	1074.6
99.9th	1518.7	1576.9	1851.9	1996.2	1982.7	1447.7	1430.8	1624.0	1829.4	1914.0
99.95th	1883.3	1974.8	2232.7	2399.8	2380.8	1832.2	1776.5	2029.7	2217.2	2313.5
99.99th	2805.5	3059.4	3263.0	3432.1	3586.1	2770.5	2697.5	3104.8	3156.2	3467.7
Std. Dev.	112.32	125.53	146.36	161.22	165.44	107.74	113.61	129.65	145.10	156.51

TABLE LXXXV:
PERCENTILES AND STANDARD DEVIATION OF AVERAGE PACKET DELAY PER USER FOR THE MINIMUM PERFORMANCE SCHEDULING ALGORITHM IN THE ITU VEHICULAR B CHANNEL WITH 16 USERS PER SECTOR USING THE "PF A" AND "PF B" METHODS TO CALCULATE THE AVERAGE RATE AND PERFECT MULTISLOT PREDICTION

Calc. Method	"PF A"					"PF B"				
C_k (kb/s) / Percentile	5	10	15	20	25	5	10	15	20	25
10th	10.8	13.8	19.7	28.3	37.2	10.5	12.6	16.1	22.6	31.2
25th	20.2	27.8	41.9	47.3	45.9	19.1	24.9	33.0	44.1	46.9
50th (median)	65.0	68.0	60.9	57.4	54.9	63.0	70.7	63.9	58.1	54.8
75th	127.2	89.2	75.4	70.7	66.0	141.4	95.7	77.5	68.4	64.5
90th	162.3	107.6	91.3	82.2	75.3	175.2	113.5	90.3	78.5	73.4
Std. Dev.	68.0	40.0	32.2	30.6	34.4	76.1	43.7	33.1	29.6	32.5

TABLE LXXXVI:
PERCENTILES AND STANDARD DEVIATION OF SLOTS ALLOCATED PER USER PER DROP FOR THE MINIMUM PERFORMANCE SCHEDULING ALGORITHM IN THE ITU VEHICULAR B CHANNEL WITH 16 USERS PER SECTOR USING THE "PF A" AND "PF B" METHODS TO CALCULATE THE AVERAGE RATE AND PERFECT MULTISLOT PREDICTION

Calc. Method	"PF A"					"PF B"				
C_k (kb/s) / Percentile	5	10	15	20	25	5	10	15	20	25
10th	228	329	351	352	365	215	321	384	397	388
25th	356	454	467	459	467	331	458	498	505	496
50th (median)	719	786	723	726	715	691	800	783	756	752
75th	1577	1627	1520	1463	1401	1486	1608	1578	1469	1420
90th	2596	2410	2541	2579	2605	2729	2422	2372	2472	2507
Std. Dev.	1041.6	858.5	911.5	972.6	973.4	1102.6	861.6	828.0	872.5	915.3

Chapter 6: Summary of Work, Conclusions, and Further Work

6.1. Summary and Conclusions

This work investigated the performance of a number of scheduling algorithms operating in the IS-856 system. Three channel models were used in the evaluation: the ITU pedestrian A channel at 3 km/h, which experiences flat fading; the ITU pedestrian B channel at 3 km/h, which is a multipath environment with resolvable components and hence undergoes small-scale frequency-selective fading; and the ITU vehicular B channel at 60 km/h, which also contains resolvable multipath components, but which experiences faster fading than the pedestrian channels due to the higher mobile speed. For all three channels, both outdated channel information plus margins and perfect multislot prediction were used to determine the bit rates and transmission formats each user in the sector could support during any given time slot. In the case of the vehicular channel, perfect one-slot prediction was also used (i.e. perfect prediction of the channel state for the first transmission slot as opposed to all possible transmission slots in the case of multislot prediction). Also for the vehicular channel, the outdated channel state was defined by the signal-to-interference ratio (SIR) with the Rayleigh component removed. This approach was used since the fading is fast enough in the vehicular channel that the current value of the SIR does not give much of an indication what the value will be when a packet transmission actually occurs (this is assumed to be three slots after the current SIR is measured). For the pedestrian channels, the fading is much slower, so the Rayleigh component was included when making format determinations.

Examining the results of the simulations, a tradeoff between throughput, delay, and fairness was observed. In general, the larger the average sector throughput an algorithm provides, the lower average delay per packet the system will see as well. This occurs since the algorithm usually ends up transmitting many packets in succession to one user or a group of users. However, the high throughput also results in a large average packet delay experienced per user. Since only a small group of users receive most of the system's resources, the other users in the system experience very long delays for their packets, and hence on average the delay per user is quite high. Also associated with the high throughput is a less fair distribution of the system resources, as just mentioned. By boosting the sector throughput by transmitting to only a small subset of users, the differences seen in the throughput and delay experienced by each user are quite substantial. There is also a large variation in the number of slots allocated to each user. In some extreme cases, users who experience the poorest channel conditions may not receive any service during a given drop at all.

If an algorithm attempts to provide a fairer distribution of resources to the users, the resulting sector throughput will drop and the average delay per packet will increase. However, the average delay per user will decrease. The distributions of the resources will not be as spread out. For example, the throughputs some users receive will no longer be as large, but the variation in the level of throughput each user receives will be much smaller. There would also usually be a far smaller probability of any particular packet or user experiencing an extremely long delay.

It should be noted, however, that there is a limit to how much an algorithm can adjust the throughput/delay tradeoff before negative effects start occurring. As an example, an algorithm may attempt to reduce the packet delay seen by users of the system. However, if the algorithm focuses excessively on the delay to the point that the channel conditions of the users start to become ignored, the average delay can actually increase, contrary to the goal of the algorithm. Essentially what ends up happening is that if delay becomes the deciding factor in the scheduling decision, the throughputs experienced by the system and the users drop dramatically, since multiuser diversity is not exploited by the system. Simultaneously, although the probability of an extremely long delay decreases, on average the delay experienced by users and by packets increases. Instead of transmitting to users in good channel conditions, the system resources are instead hoarded by the users in poor channel conditions. The packets of those latter users are sent at low bit rates and will often require several re-transmission attempts to be received correctly. During that time, the system is unavailable to transmit data to other users, and consequently the average delay experienced by all packets and by all users increases. In contrast, algorithms that consider channel conditions more strongly postpone transmissions to users in poor channel conditions until they can support higher data rates via formats that take fewer slots to transmit. As a result, the average delay experienced overall is smaller.

Examining the results for the individual algorithms, the maximum rate (maxD) algorithm provided the highest sector throughput, while the round robin (RR) algorithm usually provided the lowest. The proportionally fair (PF) algorithm sector throughput was around the middle of the range defined by the previous two algorithms. The maximum rate algorithm also displayed the lowest average delay per packet, but yielded an extremely large average delay per user, as users often went without service for an entire drop. The differences in the amount of service in terms of throughput, packet delays, and allocated slots for each user were also substantial. The round robin algorithm displayed rather large values for both the average delay per packet and the average delay per user, since the channel conditions were ignored when making scheduling decisions. However, it also displayed a fairly small variation in the throughputs, delays, and slots each user received. The proportionally fair algorithm seemed to provide a good balance between throughput, delay, and fairness. The average delay per packet was quite low, and the average delay per user was the lowest of all the algorithms investigated. The differences in the amount of service each user received were also very small.

The proposed maxD/PF hybrid algorithm performed well in accomplishing its goal of linearizing the throughput range between that of the maxD and the PF algorithms with the tuning parameter. Interestingly, the proportion of packets transmitted with each format and effective bit rate also changed approximately linearly with the tuning parameter as well. The resulting delays per packet and per user did not show a linear change, as the relationship between delay and throughput is more complex. However, there still was an apparent transition in the delays, as well as the distributions of throughputs and slots per user, as the value of the tuning parameter changed.

The M-LWDF algorithms did not perform quite as well, likely due to their overemphasizing the delay to some degree. The throughputs were generally lower than that of the PF algorithm, while the average delays experienced were higher. On the plus side, the probability of a packet experiencing an extremely long delay was quite low. The distributions of throughputs, delays, and slots also were reasonably tight. Overall, the results seen tended to be about midway between those displayed by the PF algorithm and those displayed by the RR algorithm, although the average delay per user for the M-LWDF 1 algorithm could occasionally be somewhat larger.

The slot-wise fair (slot-fair) and packet-wise fair (packet-fair) algorithms also yielded a decent performance. The slot-fair algorithm results were comparable to the PF algorithm in terms of throughput, delay, and fairness. Not surprisingly, the slot-fair algorithm showed the tightest distribution of allocated slots per user, since the goal of the algorithm was to approximately equalize the number of slots each user received. The packet-fair algorithm throughput was not very large, with values on the order of that seen for the M-LWDF algorithms. However, the throughput was still generally higher than that of the RR algorithm, which had a similar goal of providing approximately the same number of packets to each user in the sector. The average delay per packet and per user for the slot-fair algorithm was also usually much lower than that experienced under the RR algorithm. The distribution of service for the packet-fair algorithm also tended to be tighter, due to a mechanism in the scheduling metric that compensates users if they fall behind in the number of packets they receive. The round robin algorithm has no such mechanism.

The minimum performance (minperf) algorithm yielded the most interesting results out of all the algorithms analyzed. The sector throughput initially increased as the number of users in the sector increased. However, eventually the requirement of providing a specified minimum throughput to each user in the sector started to override the preference to transmit to the user with the highest requested rate. At that point, the sector throughput reached a maximum and started to decrease. However, there came a point where it was possible for it to no longer be feasible to provide the specified minimum throughput to all the users in the system. At that point, the sector throughput reached a minimum, and the system instead only provided around the minimum specified throughput to the subset of users in the best channel conditions. Meanwhile, the users in the worst channel conditions were ignored for long periods of time, occasionally for the entire drop of 30 seconds. The transition from transmitting to the best user to providing minimum throughputs could be seen in the average delay per packet and per user curves as well. The delays would initially increase at a certain rate while the best user was the algorithm's priority. However, once the algorithm started focusing more on providing minimum throughputs, the rate at which the average delay increased dropped noticeably. Considering the throughput received by each user, there was a sharp jump in the distribution around the minimum specified throughput, and a long tail at the upper end of the distribution extending into higher throughputs. This indicated that if a user could support a throughput much higher than the minimum, that user would receive that high throughput. Otherwise, a user would only receive approximately the minimum specified throughput. In the cases where providing the minimum to all users became unfeasible, there was an increase in the number of users receiving below the minimum, the distribution increased much less sharply around the minimum, and the upper tail did not extend very far into higher throughputs. These two groups of users (one well above the minimum throughput, the other around the minimum throughput) could be seen again in the distribution of packet delays seen by users. There was an initial rise in the distribution from the high throughput users, a flattening of the rate of increase, then a second rise to the group receiving around the minimum throughput. Because of this effective grouping of users into two classes, the distributions of delays per packet and slots per user became rather spread out.

The exponential rule (EXP) showed two vastly different performances depending on whether slots or seconds were used as the units for the delay in the algorithm metric. In the case of slots, the resulting throughput and delays were not much better than those experienced under the RR algorithm. Clearly, in that case, the algorithm ended up placing far too much emphasis on the delays and not enough on the requested rates. The case of seconds was a much different story. The resulting throughputs were generally only a negligible amount smaller than the throughputs of the maxD (in the case of the EXP1B algorithm) and the PF (in the EXP2B case) algorithms. The average delay per packet, throughputs per user, and slot allocations also did not change by an appreciable amount; the average delay per packet became slightly larger and the distributions became a bit narrower. The major change came in the form of the average delay per user. Although there was very little change to be seen between the EXP2B and PF algorithms, there was a dramatic reduction in the delay between the EXP1B and the maxD algorithms. In certain cases, the average delay per user decreased by about an order of magnitude. The EXP algorithm essentially ensures that every packet will be served if its delay becomes sufficiently large. The dramatic reduction in delay stems from users who received no service under the maxD algorithm now having packets sent to them. Thus, the EXP algorithm virtually removes the probability of not receiving any service during a given drop. The throughput a user receives might not be very large, but it will be greater than zero in most cases. The EXP algorithm also dramatically reduced the probability of a packet experiencing an extremely lengthy delay. This reduction could be seen for both the EXP1B and EXP2B algorithms.

The EXP extension to the maxD/PF algorithm to make the maxD/PF/EXP algorithm provided similar results. There generally was not a significant change to be seen in the throughputs, average delay per packet, or slot allocations between the maxD/PF/EXP algorithm and the respective maxD/PF counterpart.

The major change was again a noticeable reduction in the probability of an extremely lengthy packet delay. There was also a lessening to be seen in the average delay per user, but the difference became less significant as the value of the tuning parameter increased. In fact, above a tuning parameter value of 0.25 in the pedestrian channels and 0.5 in the vehicular channel, the difference was essentially negligible, as it was with the EXP2B algorithm (which is actually a special case of the maxD/PF/EXP algorithm with a tuning parameter value of 1).

Considering the change between outdated channel information and perfect prediction in the pedestrian channels, the resulting changes were varied. The difference seemed to be in how much an algorithm emphasized channel conditions in its scheduling metric and hence how many packets were transmitted to users with the lowest rate formats. For those algorithms that saw a non-negligible proportion of packets transmitted with those formats (i.e. above a few percent in total), the resulting throughput actually decreased when using perfect prediction. The reasons for this are twofold; first, when margins are used, the magnitudes of the margins for the lowest rate formats were generally fairly large. This meant that users would more often appear to not be able to support any transmission format. Second, the packets that were sent with the lowest rates often ended early when margins were used. This was no longer the case with perfect prediction, as the users would request the higher rate in the first place. Therefore, under perfect prediction, for certain algorithms, users were deemed to be capable of receiving low transmission rates more often, and when they received packets under those formats, the transmissions did not terminate very early. As a consequence, the sector throughput decreased for those algorithms. The delays experienced by packet and by users also tended to increase for the same reasons. For the algorithms which sent few or no packets at the lowest rates, perfect prediction did cause an improvement in the throughput, usually with the associated decrease in average delay per packet, increase in the average delay per user, and widening of the various distributions. Therefore, it was seen that the use of margins could be an effective replacement for prediction, and in some cases could actually yield a better performance than perfect prediction would.

It was also observed that the fluctuations in the signal strength for each user are a necessary factor in order to obtain a reasonable system performance and yield a decent amount of multiuser diversity. It was seen that using the SIR_p values (with the Rayleigh component of the signal removed / averaged out) in the vehicular channel resulted in very poor throughputs. In many cases there was little or no gain in the sector throughput as the number of users per sector increased. The resulting delays were also extremely bad, with a much larger probability of being in outage for long periods and/or not receiving any service during an entire drop. The effective channel capacity was reduced as a result of not having an accurate measure of the signal strengths. There was a dramatic improvement to be had when using perfect one-slot prediction, as the channel strength variations could become accurately tracked. In fact, the resulting throughput was even larger than in the ITU pedestrian B channel with perfect multislot prediction. Therefore, it was shown that the fluctuations in the signal strength are required in order to obtain a decent level of performance. The relative performance of the ITU vehicular B channel over the ITU pedestrian B channel would also seem to indicate that larger and more rapid changes are better, providing that they can be tracked with enough precision. If not, the changes result in a deterioration of the system performance.

Moving from one-slot prediction to multislot prediction in the vehicular channel showed an improvement in throughput at low numbers of users per sector, as users with low to intermediate signal strengths were able to request higher rates. However, at higher numbers of users per sector, the throughput decreased when using multislot prediction. This is due to the fact that negative margins were used with one-slot prediction to give the two highest rate formats a 1% packet error rate. The negative margins also resulted in those formats being chosen more often than with multislot prediction, where no margins were used. Additionally, those algorithms that transmit often with low rate formats saw a further reduction in throughput, in a similar manner and for the same reasons as discussed above for the switch from outdated channel information to perfect multislot prediction in the pedestrian channels.

A new method of calculating the average rate in the metric of several of the algorithms (primarily the PF algorithm) was introduced in this work. The new method takes into account early terminations in the IS-856 system, which the method most commonly known and used with the PF algorithm does not. This new calculation method ("PF B") was tested in all the channels and for all the prediction methods. In most cases, there was no significant change in any of the results for the algorithms. This was likely due to two factors. First, for some algorithms, very few packets are sent with low data rate formats. Therefore, there are few opportunities for early terminations, and so the calculation methods would not yield significantly different values. Second, in the cases where early terminations were more common, it was likely that the average rate value increased by approximately the same proportion for all users. Since most of the algorithms involve a division by the average rate in the metric, the user who had the relatively largest metric would not change, and so the same users would be selected, and hence the results would not change. The major exception to all of this was the minperf algorithm. The metric for the minperf algorithm involved a subtraction of the average rate instead of a division. As such, the relatively best user did change, and there was a significant difference in the simulation results. There was a marked increase in throughput, as the more accurate calculation method allowed users to operate closer to the actual minimum specified throughput. This freed up system resources to allocate to the users requesting the highest rates. Associated with the higher throughput were the expected decrease in average delay per packet, increase in the average delay per user, and a widening of the various distributions. It should be noted, however, that in many cases the change in the average delays was only minor. One other key observation was that there were significant changes for all the algorithms with an average rate calculation (not just the minperf algorithm) in the vehicular channel when outdated SIR_p values were used to determine the supportable bit rates. In that scenario, early terminations were quite common for all the algorithms and all the formats, and hence the calculation method that accounted for those early termination made a noticeable difference. The changes seen for all the algorithms were similar to what the minperf algorithm experienced in all other cases when the "PF B" method was used (i.e. increased throughput, decreased average delay per packet, and so on). In general, the "PF B" method of calculating the average rate is probably superior and a better choice when there is even a slight chance of early terminations. In many cases, the difference might be minimal, but in some cases, there is a very dramatic improvement over the original calculation method.

6.2. Contributions and Future Work

This work has presented a substantial analysis and comparison of many different scheduling algorithms. This in and of itself is a useful result. In general, whenever a new scheduling algorithm is presented in the literature, there will be some sort of accompanying analysis of how that algorithm performs relative to a few well-known algorithms. However, those analyses rarely use the same criteria for testing (i.e. channel models, traffic models, etc.) Therefore, a comprehensive comparison of these algorithms all under the same conditions is a worthwhile endeavor.

Besides the comprehensive comparison, several new algorithms or modifications of existing algorithms were also proposed in this work. Primarily, the maxD/PF algorithm was created and evaluated. It was shown that the algorithm performed well in its goal of providing a tunable throughput within the range of that provided by the maxD and PF algorithms. The change in throughput was almost always approximately linear with the tuning parameter, except in the case when reliable channel estimates were not available. An alteration to the slot-fair algorithm was also proposed, so that the shared resource was no longer time (i.e. slots), but packets instead. This thereby created the packet-fair algorithm, which could be more directly compared to the round robin algorithm. It was seen that the packet-fair algorithm almost always performed much better relative to the RR algorithm. The one exception was again in the case when reliable channel estimates were unavailable. Other alterations to algorithms that were examined included the effects when the channel capacity was not checked ahead of time for the minperf algorithm, and the effects of using different units of measure for the head-of-line delay for the EXP algorithm.

As part of the analysis of the proportionally fair algorithm, a new method was proposed to calculate the average rate in the metric of the algorithm. The method most commonly used does not take into account the early terminations that can occur within the 1xEV-DO system, whereas the proposed method does with no change in complexity. It was seen that the new calculation method in most cases only increased the throughput of the PF algorithm and other algorithms that involved a division by the average rate in their metric by a negligible amount. The delay and fairness results were similarly unchanged. However, for the minperf algorithm, which involved a subtraction from the average rate rather than a division, there was a marked increase in the throughput, with an associated decrease in the average delay per packet and increase in the average delay per user. In addition, in the ITU vehicular B channel, when outdated channel information was used, all the algorithms showed an increase in throughput, due to the large numbers of packets received with early terminations. Therefore, the cases when there was a discernible improvement suggest that the new method should be used in all cases. In many cases when using the proposed calculation method, little performance is gained, but in some cases, the performance gain is large, and in all cases, no performance is lost.

The comparison of the methods to determine the format each user could support showed that the use of outdated channel information and margins made it less likely for users to receive packets with the lowest rate formats, and when those formats were used, early terminations were often seen. Depending on the overall proportion of packets received with those formats, the throughput of the algorithm could increase or decrease compared to the case when perfect multislot prediction was used. The algorithms that transmitted packets the most often with the lower rate formats showed higher throughput with the use of margins. Therefore, it was shown in this work that the use of outdated channel information and margins can be an effective alternative to prediction, and in some cases can result in a better system performance.

Future work on this topic might include seeing what the effect of a combined pedestrian/vehicular channel might have on the throughput, delay, and fairness. That is, within a sector, some of the users would be in a pedestrian environment, while others would be in a vehicular environment. Using individual margins for each user and adjusting them independently might be an improvement on the simulations. It would also be worthwhile to use an actual packet data traffic model (e.g. web browsing, real-time streaming, etc.) to generate incoming packets instead of assuming that all users have an infinite backlog of packets. If certain users go into dormant periods during which they do not request any data, it is possible that an algorithm like maxD may provide a sufficient amount of service to all users in the sector.

During this work, it was assumed that the type of traffic being sent was highly delay tolerant, such as web browsing or e-mail services. Future work on this topic could extend the analysis of these algorithms to the case when the traffic is not quite so tolerant to delays. An example of this might be the real-time streaming of an audio or video file. Associated with the traffic type and model might be certain Quality of Service measures that should be met such as a minimum throughput or maximum delay, and which could be different for each user. The algorithms could be tested to see how well they meet those measures. In such a case, more individualized statistics such as the differences in the delay experienced by the packets of an individual user (i.e. the delay jitter) could be examined.

Bibliography

[1] TIA/EIA-95-B, *Mobile Station-Base Station Compatibility Standard for Wideband Spread Spectrum Cellular Systems*, Telecommunications Industry Association, Arlington, VA, Feb. 1999.

[2] TIA/EIA/IS-2000.2, *Physical Layer Standard for cdma2000 Spread Spectrum Systems*, Telecommunications Industry Association, Arlington, VA, August 1999.

[3] R. Knopp and P. A. Humblet, "Information capacity and power control in single-cell multiuser communications," in *Proc. IEEE ICC'95*, Seattle, WA, June 1995, vol. 1, pp. 331-335.

[4] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad-hoc wireless networks," in *Proc. IEEE INFOCOM 2001*, Anchorage, AK, Apr. 2001, vol. 3, pp. 1360-1369.

[5] A. Bedekar, S. Borst, K. Ramanan, P. Whiting, and E. Yeh, "Downlink scheduling in CDMA data networks," in *Proc. IEEE GLOBECOM'99*, Rio de Janeireo, Brazil, Dec. 1999, vol. 5, pp. 2653-2657.

[6] TIA/EIA/IS-856, *cdma2000 High Rate Packet Data Air Interface Specification*, Telecommunications Industry Association, Arlington, VA, Nov. 2000.

[7] TIA/EIA/IS-2000.2, *Physical Layer Standard for cdma2000 Spread Spectrum Systems – Release C*, Telecommunications Industry Association, Arlington, VA, May 2002.

[8] 3GPP TS 25.308 V5.4.0 (2003-03), "UTRA High Speed Downlink Packet Access (HSDPA); Overall Description; Stage 2 (Release 5)", Third Generation Partnership Project, Mar. 2000.

[9] S. Parkvall, E. Dahlman, P. Frenger, P. Beming, and M. Persson, "The high speed packet data evolution of WCDMA," in *Proc. IEEE International Symposium on Pers., Indoor, and Mobile Radio Commun. (PIMRC)*, San Diego, CA, Sept.-Oct. 2001, vol. 2, pp. G-27–G-31.

[10] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana, and A. Viterbi, "CDMA/HDR: a bandwidth-efficient high-speed wireless data service for nomadic users", *IEEE Commun. Mag.*, vol. 38, no. 7, pp. 70-77, July 2000.

[11] D. W. Paranchych and M. Yavuz, "A method for outer loop rate control in high data rate wireless networks," in *Proc. IEEE Vehicular Technology Conf. (VTC 2002-Fall)*, Vancouver, Canada, Sept. 2002, vol. 3, pp. 1701-1705.

[12] M. Yavuz and D. W. Paranchych, "Adaptive rate control in high data rate wireless networks," in *Proc. IEEE Wireless Commun. and Networking Conf.* (WCNC 2003), New Orleans, LA, Mar. 2003, vol. 2, pp. 866-871.

[13] Y. Cao and V. O. K. Li, "Scheduling algorithms in broad-band wireless networks," *Proc. IEEE*, vol. 89, no. 1, pp. 76-87, Jan. 2001.

[14] F. Kelly, "Charging and rate control for elastic traffic," *European Trans. on Telecommunications*, vol. 8, pp. 33-37, 1997.

[15] David Tse, "Multiuser diversity in wireless networks," presentation at Stanford University, Apr. 16, 2001.

[16] A. Jalali, R. Padovani, and R. Pankaj, "Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system," in *Proc. IEEE Vehicular Technology Conf. (VTC 2000 Spring)*, Tokyo, Japan, May 2000, vol. 3, pp. 1854-1858.

[17] J. Holtzman, "CDMA Forward Link Waterfilling Power Control," in *Proc. IEEE Vehicular Technology Conf. (VTC 2000 Spring)*, Tokyo, Japan, May 2000, vol. 3, pp. 1663-1667.

[18] J. M. Holtzman, "Asymptotic analysis of proportional fair algorithm," in *Proc. IEEE International Symposium on Pers., Indoor, and Mobile Radio Commun. (PIMRC)*, San Diego, CA, Sept.-Oct. 2001, vol. 2, pp. F-33–F-37.

[19] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, "Providing quality of service over a shared wireless link," *IEEE Commun. Mag.*, vol. 39, no. 2, pp. 150-154, Feb. 2001.

[20] X. Liu, E. K. P. Chong, and N. B. Shroff, "Opportunistic transmission scheduling with resource-sharing constraints in wireless networks," *IEEE J. Select. Areas Commun.*, vol. 19, no. 10, pp. 2053-2064, Oct. 2001.

[21] X. Liu, E. K. P. Chong, and N. B. Shroff, "Transmission scheduling for efficient wireless resource utilization with minimum-performance guarantees," in *Proc. IEEE Vehicular Tech. Conf. (VTC 2001 Fall)*, Atlantic City, NJ, Oct. 2001, vol. 2, pp. 824-828.

[22] X. Liu, E. Chong, and N. Shroff, "A framework for opportunistic scheduling in wireless networks," *Computer Networks*, vol. 41, no. 4, pp. 451-474, Mar. 15, 2003.

[23] S. Shakkottai and A. Stolyar, "A study of scheduling algorithms for a mixture of real and non-real time data in HDR," *Bell Labs Tech. Memo*, Oct. 2000.

[24] T. S. Rappaport, *Wireless Communication: Principles and Practice*. New Jersey: Prentice Hall PTR, 1996.

[25] ETSI TR 101 112 V3.2.0 (1998-04), "Universal Mobile Telecomunnications System (UMTS); Selection procedures for the choice of radio transmission technologies of the UMTS (UMTS 30.03 version 3.2.0)," European Telecommunications Standards Institute, Apr. 1998.

[26] G. L. Stüber, *Principles of Mobile Communication, 2nd ed.* Boston/Dordrecht: Kluwer Academic Publishers, 2001.

[27] M. Thiessen, "Determining the cell radius in an urban environment," Ericsson internal report, 1998.

[28] R. Love, R. Nikides, and L. Jalloul, "Evaluation methods for 1xEV-DV: Part I Forward Link," Motorola 3GPP2 recommendation, C50-Eval-20001005-001, Oct. 2000.

[29] R. Love, R. Nikides, and L. Jalloul, "Evaluation methods for high speed forward link packet access for 1XTREME," Motorola 3GPP2 recommendation, 3GPP2-C00-1XTR_20000713-004.REV2, Aug. 2000.

[30] M. Gudmindson, "Correlation model for shadow fading in mobile radio systems," Electronics Letters, vol. 27, no. 23, pp. 2145-2146, Nov. 7, 1991.

[31] Rec. ITU-R M.1255, "Guidelines for evaluation of radio transmission technologies for IMT-2000," International Telecommunication Union, Feb. 1997.

[32] J. Proakis, *Digital Communications, 3rd Ed.* New York: McGraw-Hill, 1995.

[33] R. H. Clarke, "A statistical theory of mobile-radio reception," *Bell Systems Tech. J.*, Vol. 47, pp. 957-1000, 1968.

[34] D. J. Young and N. C. Beaulieu, "The generation of correlated Rayleigh random variates by inverse discrete Fourier transform," *IEEE Trans. Commun.*, vol. 48, no. 7, pp. 1114-1127, July 2000.

[35] V. Vanghi, Ericsson Wireless Communications, San Diego, CA, private communication, Oct. 2000.

[36] Y. Wei, E. Tiedemann, J. Holtzman, and T. Chen, "Justification of maximum C/I for forward link," Qualcomm 3GPP2 recommendation, 3GPP2-C50-20001016-001, Oct. 2000.

[37] Y. Wei and T. Chen, "Preliminary evaluation methodology for cdma2000 1xEV-DV proposals," Qualcomm 3GPP2 recommendation, 3GPP2-C50-20001005-001, Oct. 2000.

[38] A. Duel-Hallen, S. Hu, and H. Hallen, "Long-range prediction of fading signals," *IEEE Signal Processing Mag.*, vol. 17, no. 3, pp. 62-75, May 2000.

[39] Michel C. Jeruchim, Philip Balaban, and K. Sam Shanmugan, *Simulation of Communication Systems: Modeling, Methodology, and Techniques* (2^{nd} *ed.*). New York: Kluwer Academic / Plenum Publishers, 2000.

Appendix A: AWGN Packet Error Rate Curves for 1xEV-DO

This appendix contains packet error rates curves for all possible transmission formats and effective bit rates for the 1xEV-DO system in the AWGN channel.

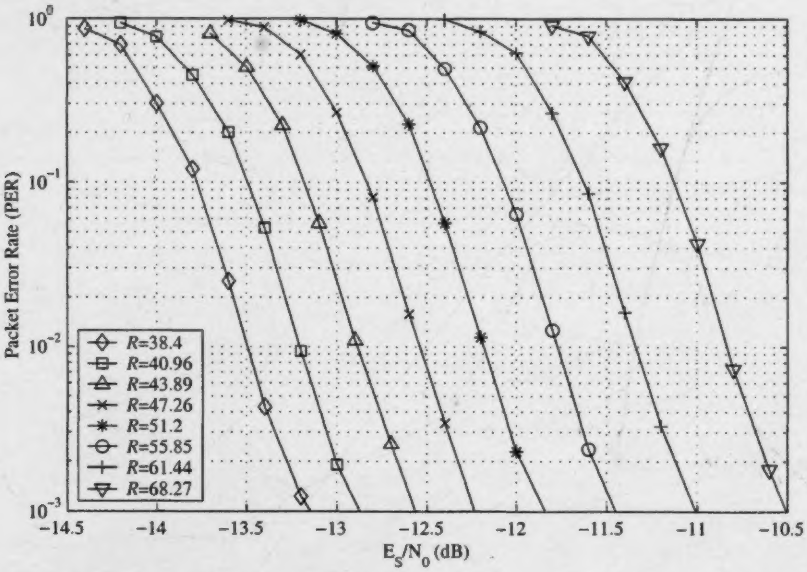


Figure A.1: PER performance in an AWGN channel for an *R* kb/s transmission of a 1024-bit packet using a maximum of 16 slots (1 of 2)

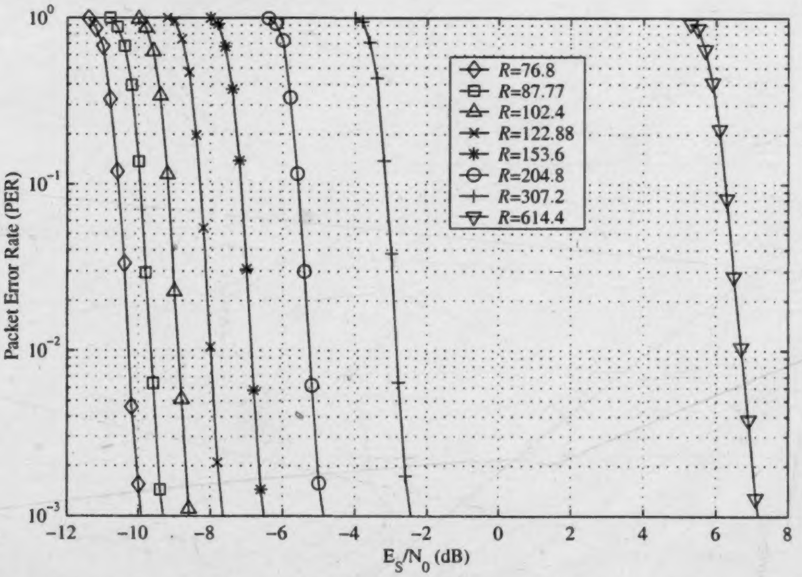


Figure A.2: PER performance in an AWGN channel for an *R* kb/s transmission of a 1024-bit packet using a maximum of 16 slots (2 of 2)

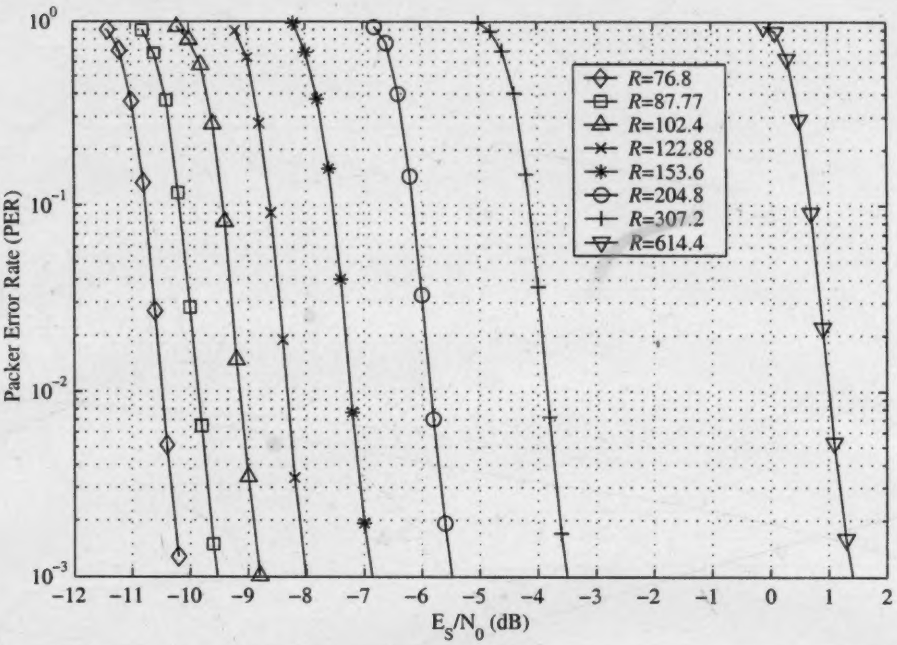


Figure A.3: PER performance in an AWGN channel for an R kb/s transmission of a 1024-bit packet using a maximum of 8 slots

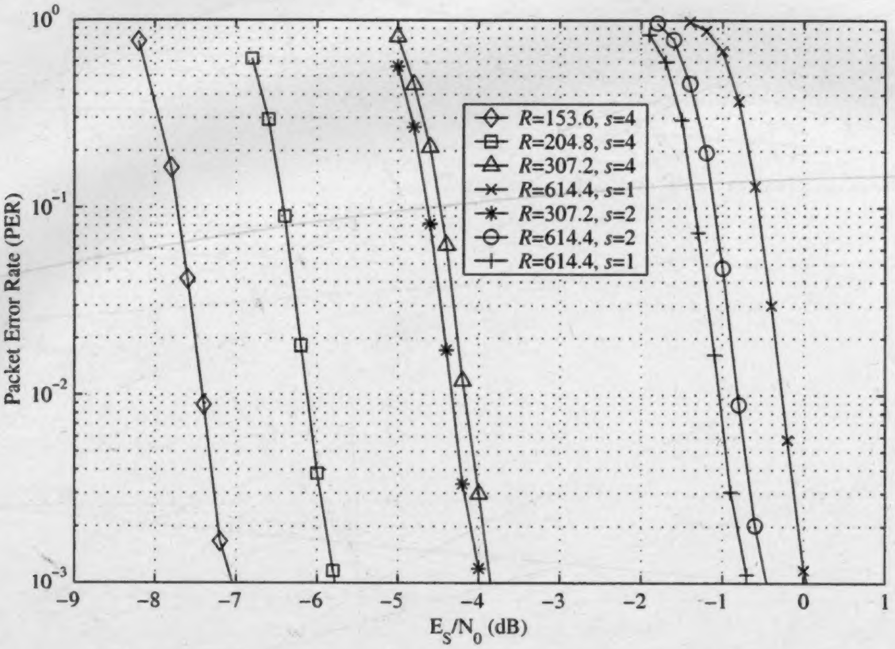


Figure A.4: PER performance in an AWGN channel for an R kb/s transmission of a 1024-bit packet using a maximum of s slots

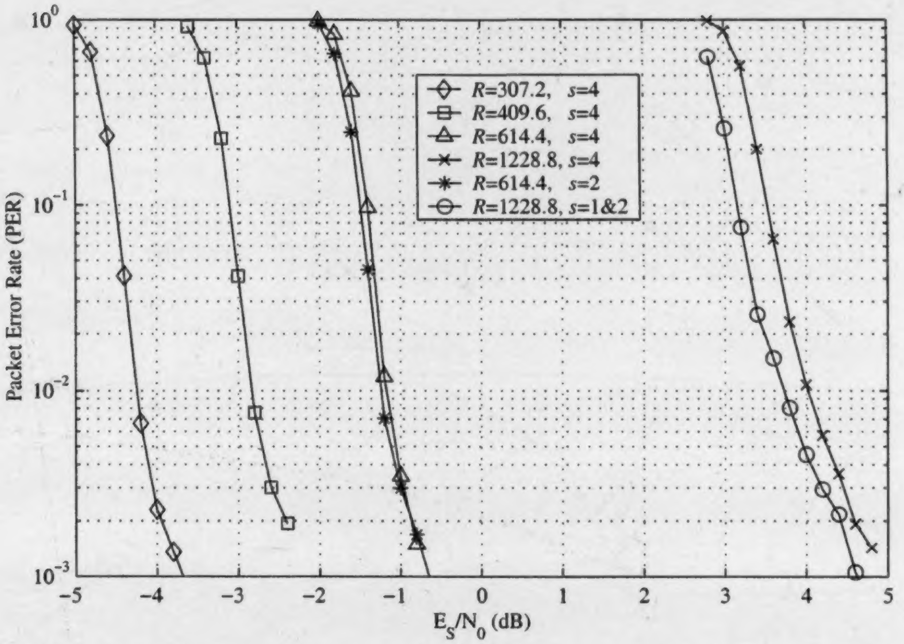


Figure A.5: PER performance in an AWGN channel for an R kb/s transmission of a 2048-bit packet using a maximum of s slots

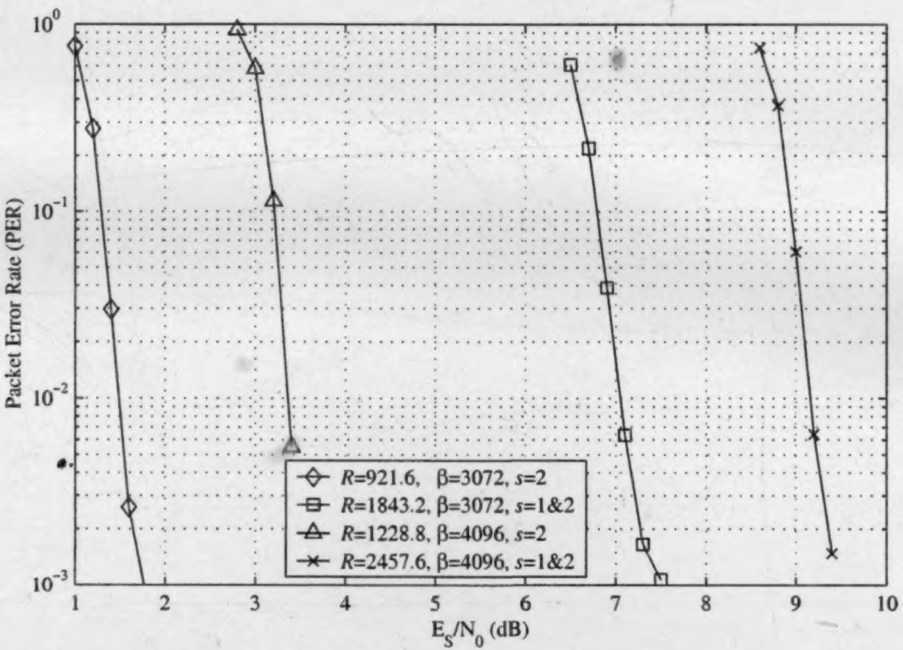


Figure A.6: PER performance in an AWGN channel for an R kb/s transmission of a β-bit packet using a maximum of s slots

Appendix B: Pedestrian Channel Results using Outdated Channel Information and Margins

This appendix contains graphs showing the simulation results for the ITU pedestrian A and B channels when outdated channel information plus margins were used to determine the transmission format each user could support.

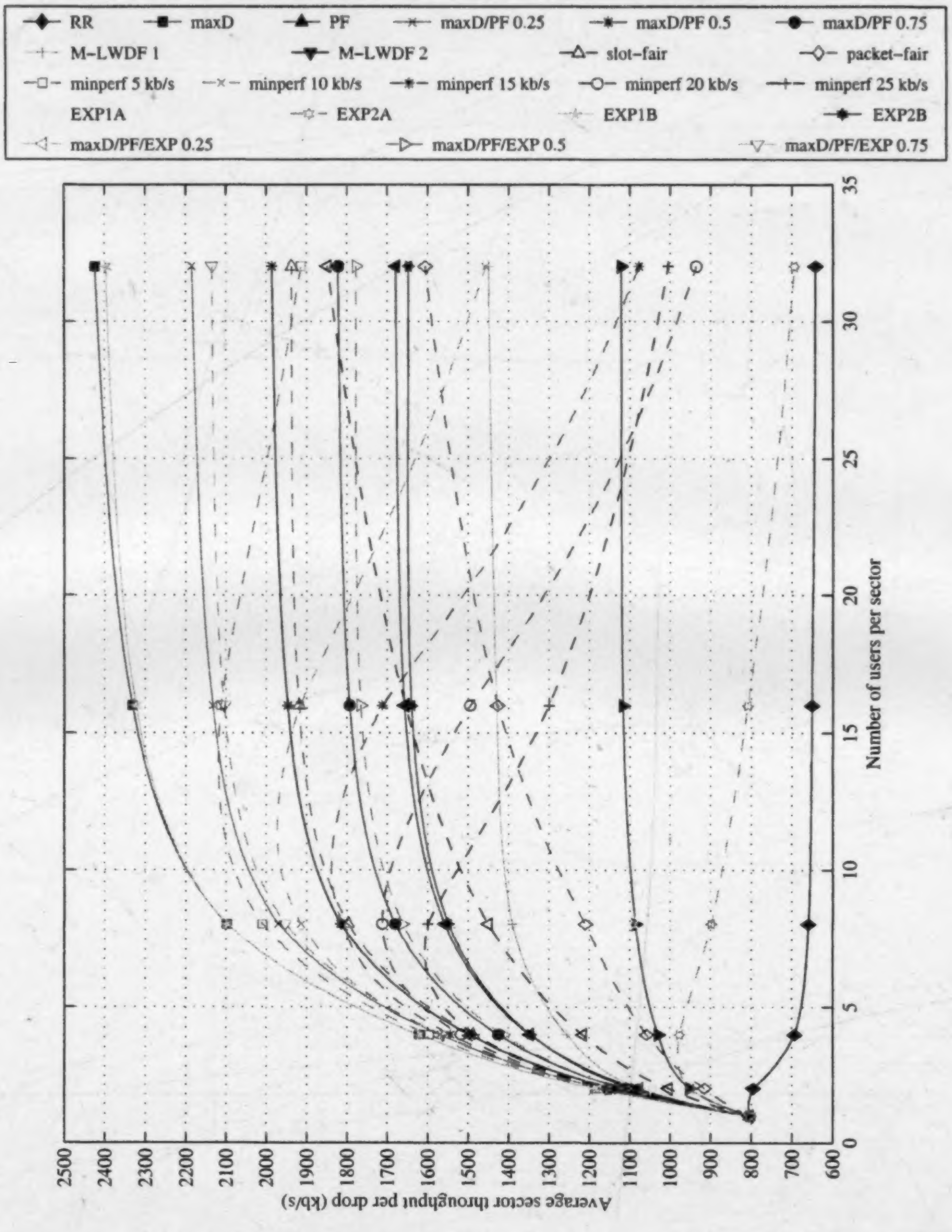


Figure B.1: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian A channel

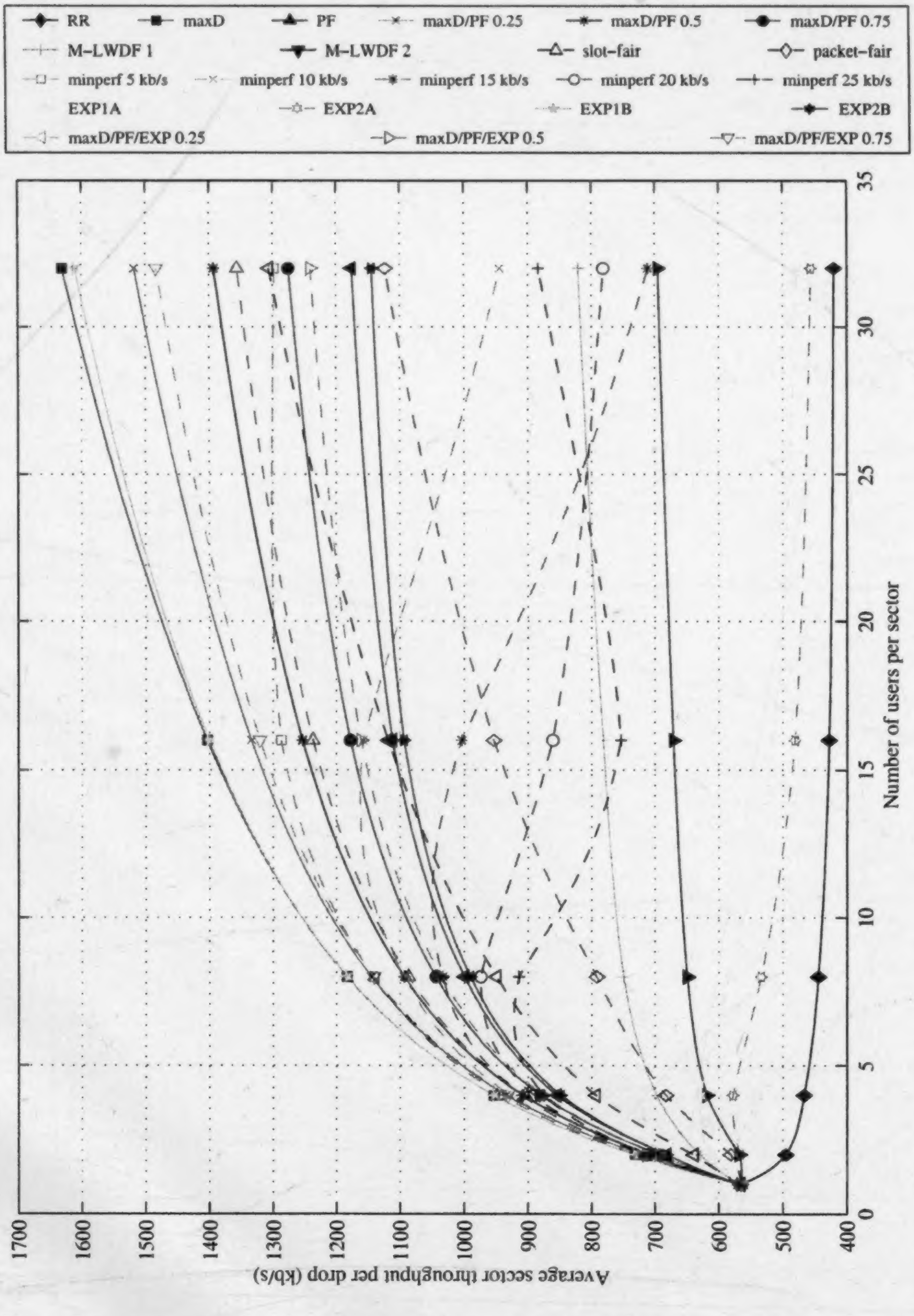


Figure B.2: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian B channel

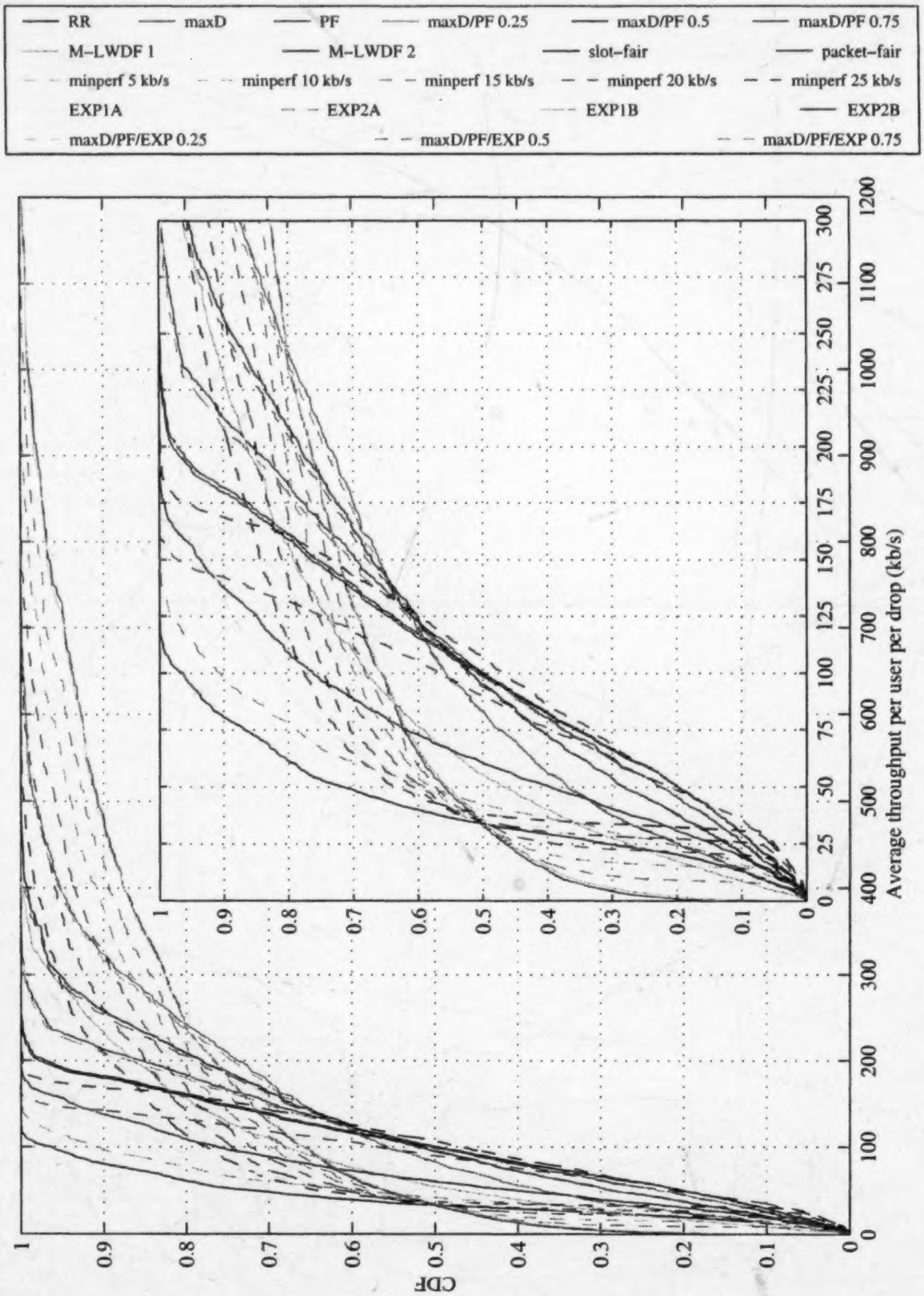


Figure B.3: Distributions of average throughput per user per drop in the ITU pedestrian A channel with 16 users per sector

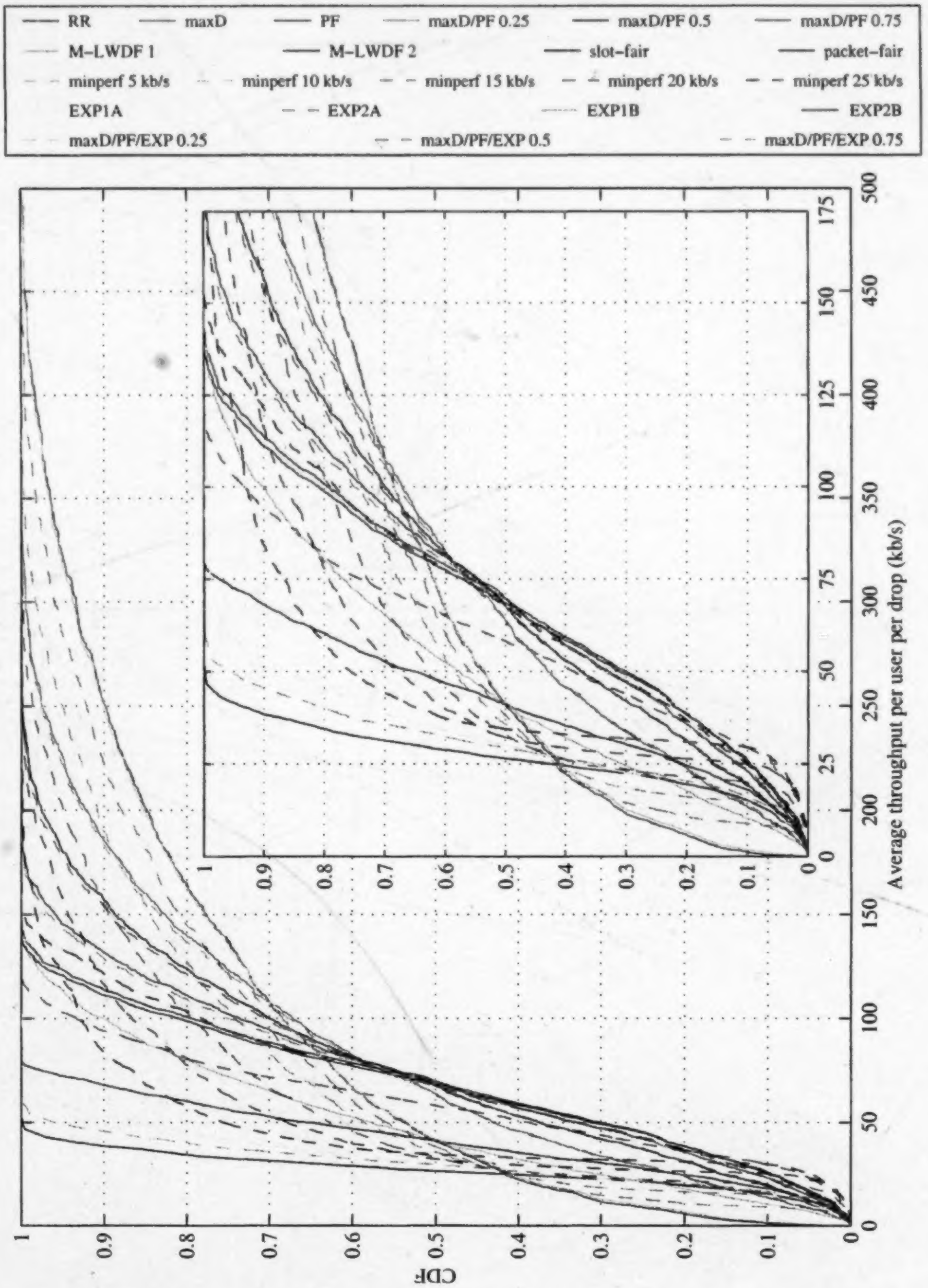


Figure B.4: Distributions of average throughput per user per drop in the ITU pedestrian B channel with 16 users per sector

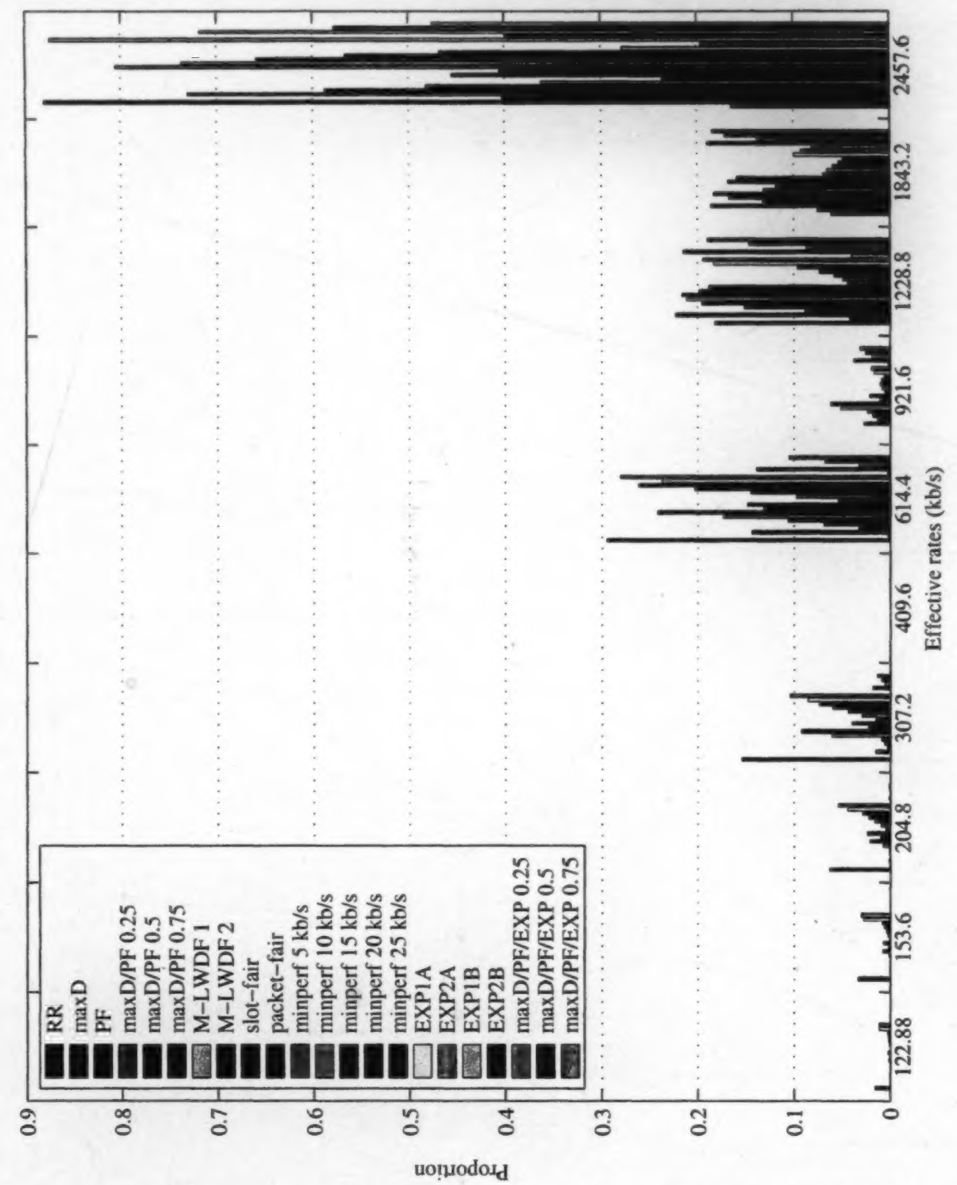


Figure B.5: Distribution of effective bit rates for transmitted packets in the ITU pedestrian A channel with 16 users per sector

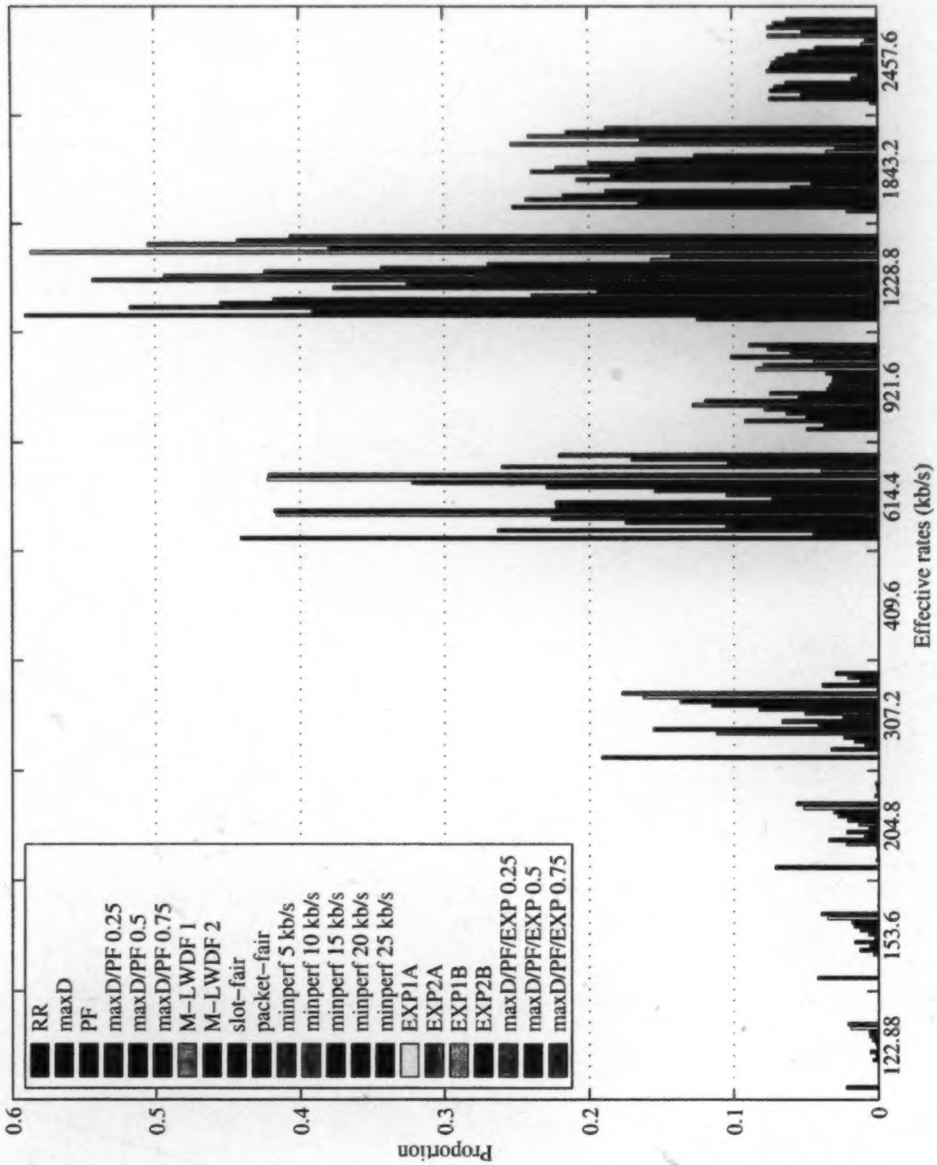


Figure B.6: Distribution of effective bit rates for transmitted packets in the ITU pedestrian B channel with 16 users per sector

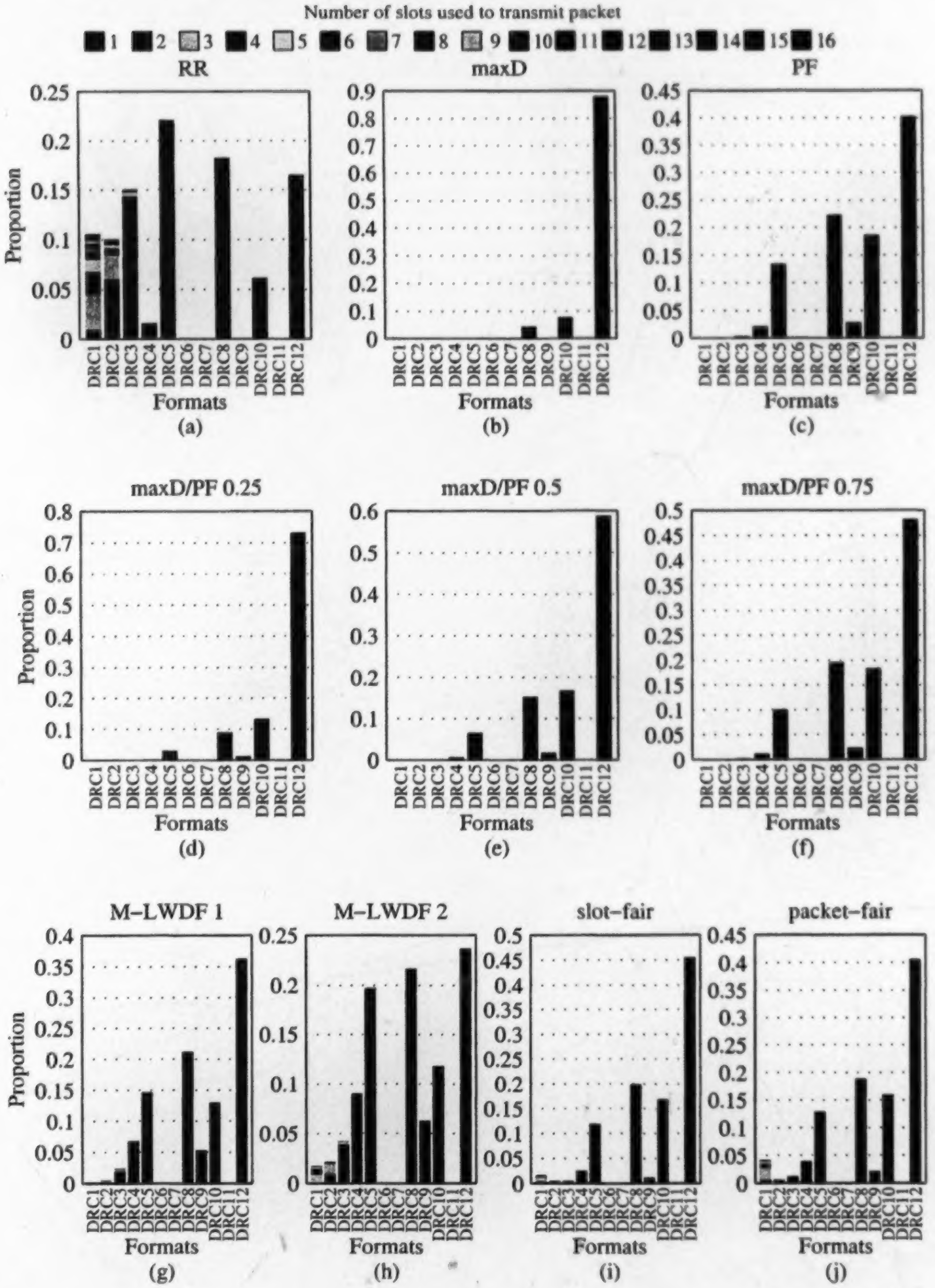


Figure B.7 (Part 1 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian A channel with 16 users per sector

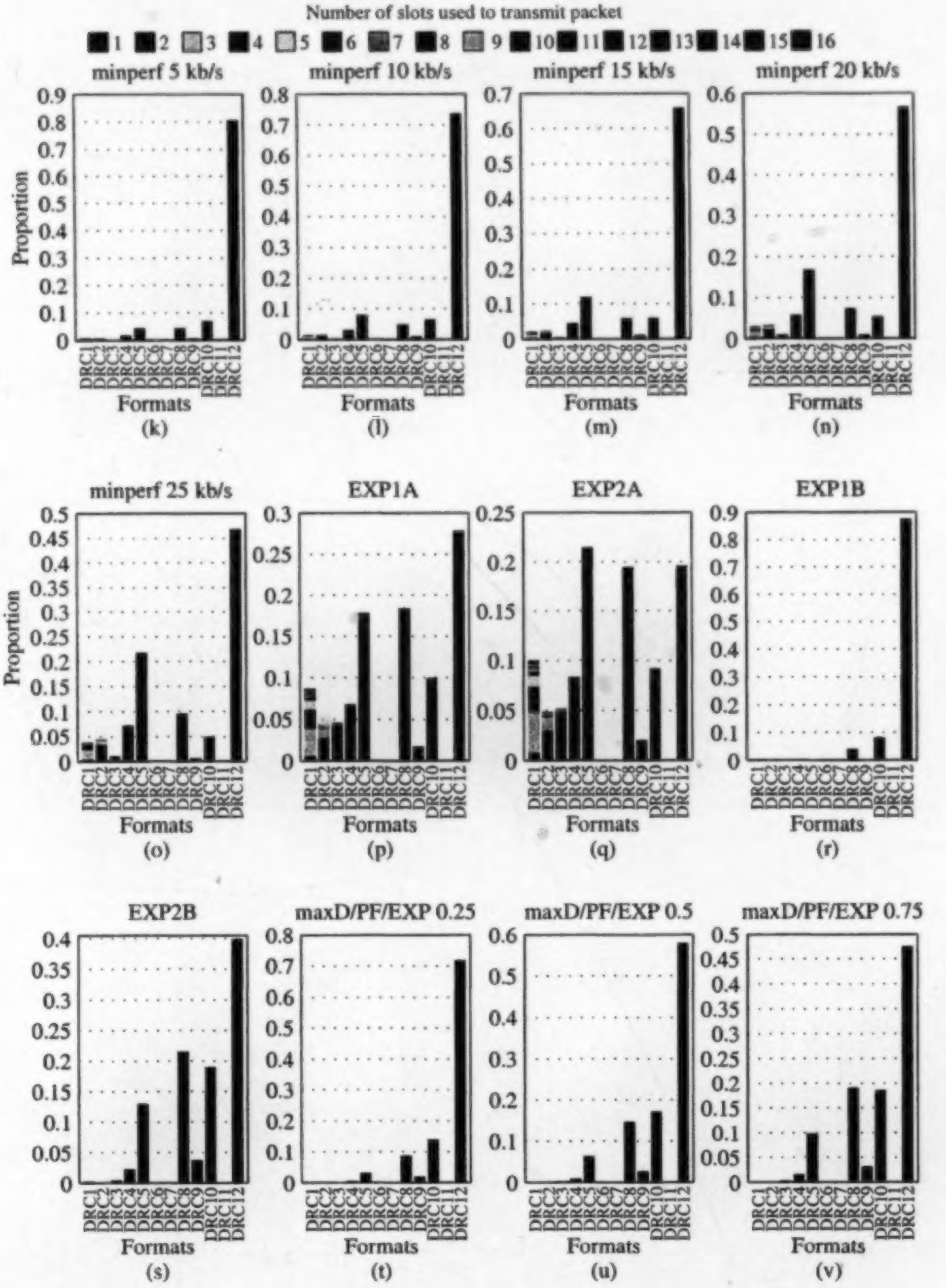


Figure B.7 (Part 2 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian A channel with 16 users per sector

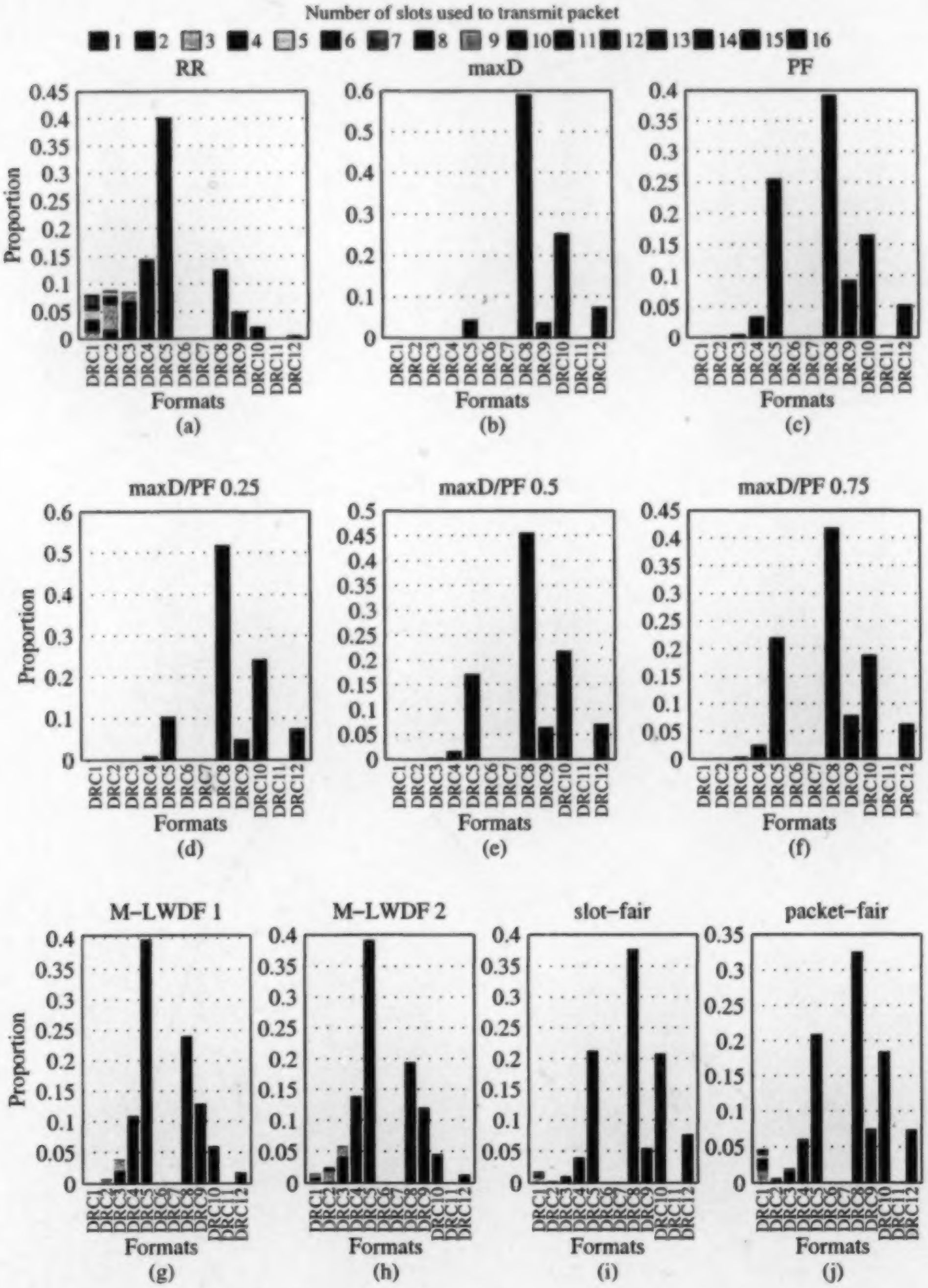


Figure B.8 (Part 1 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian B channel with 16 users per sector

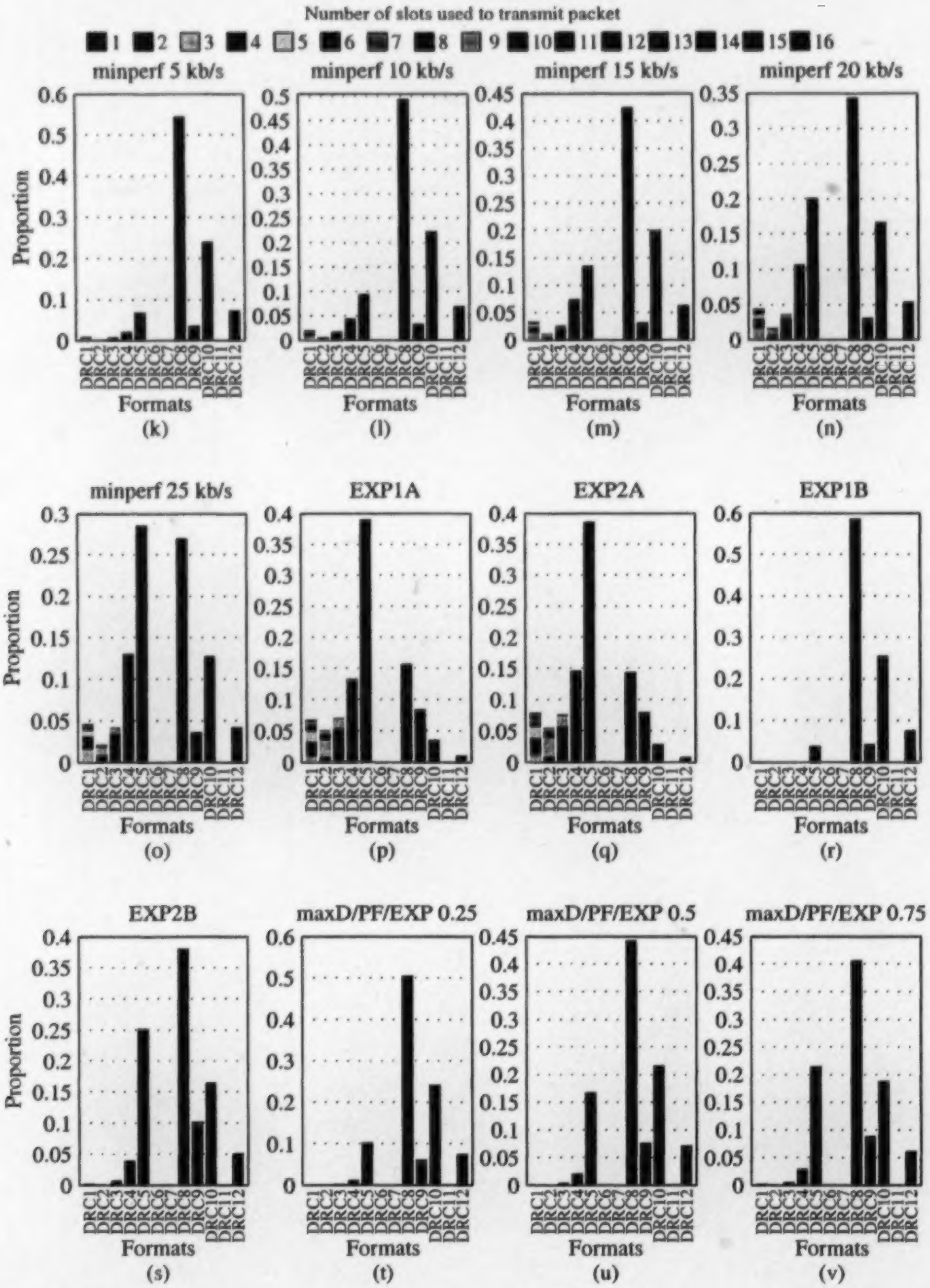


Figure B.8 (Part 2 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian B channel with 16 users per sector

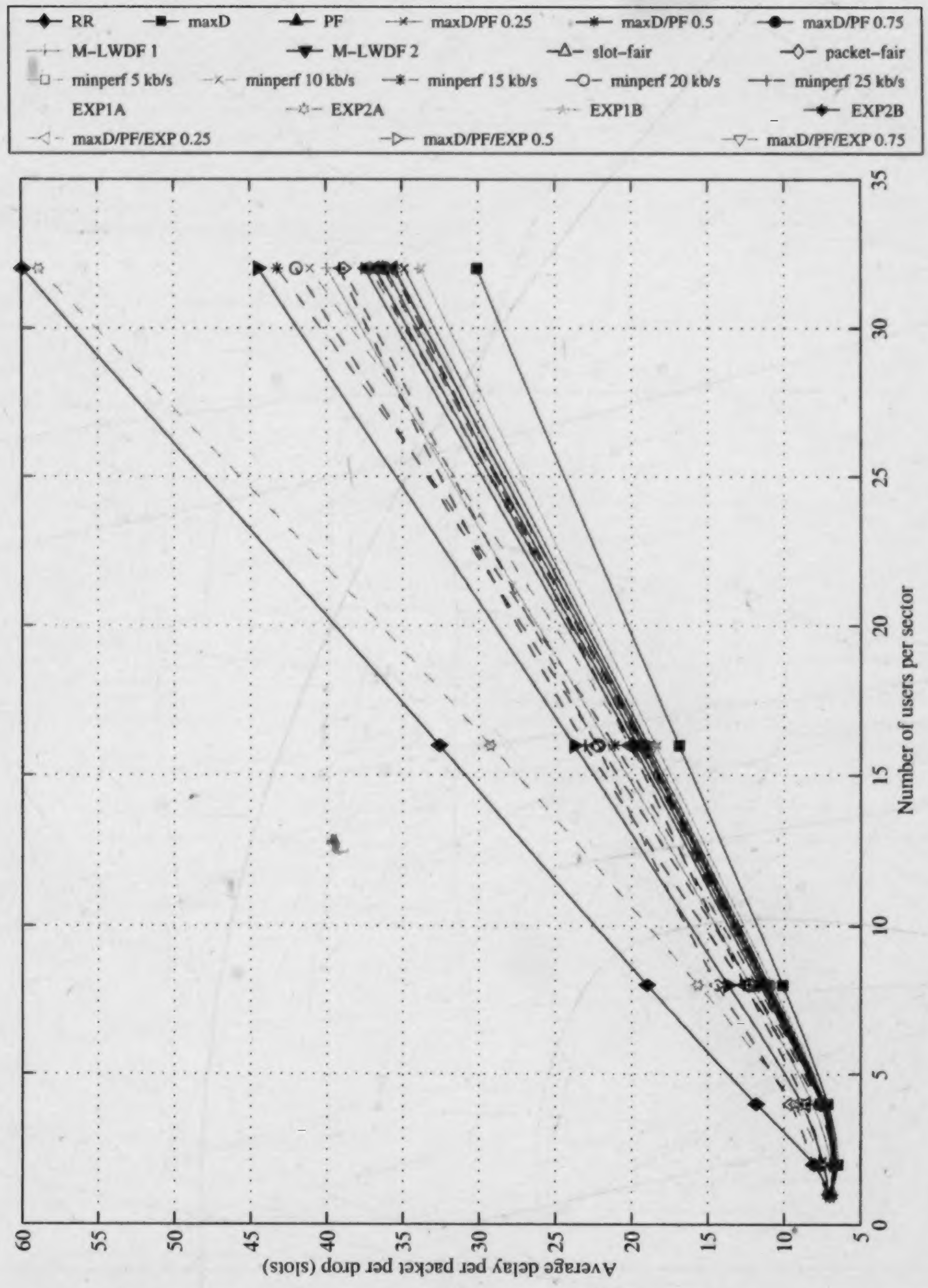


Figure B.9: Average delay per packet per drop versus number of users per sector in the ITU pedestrian A channel

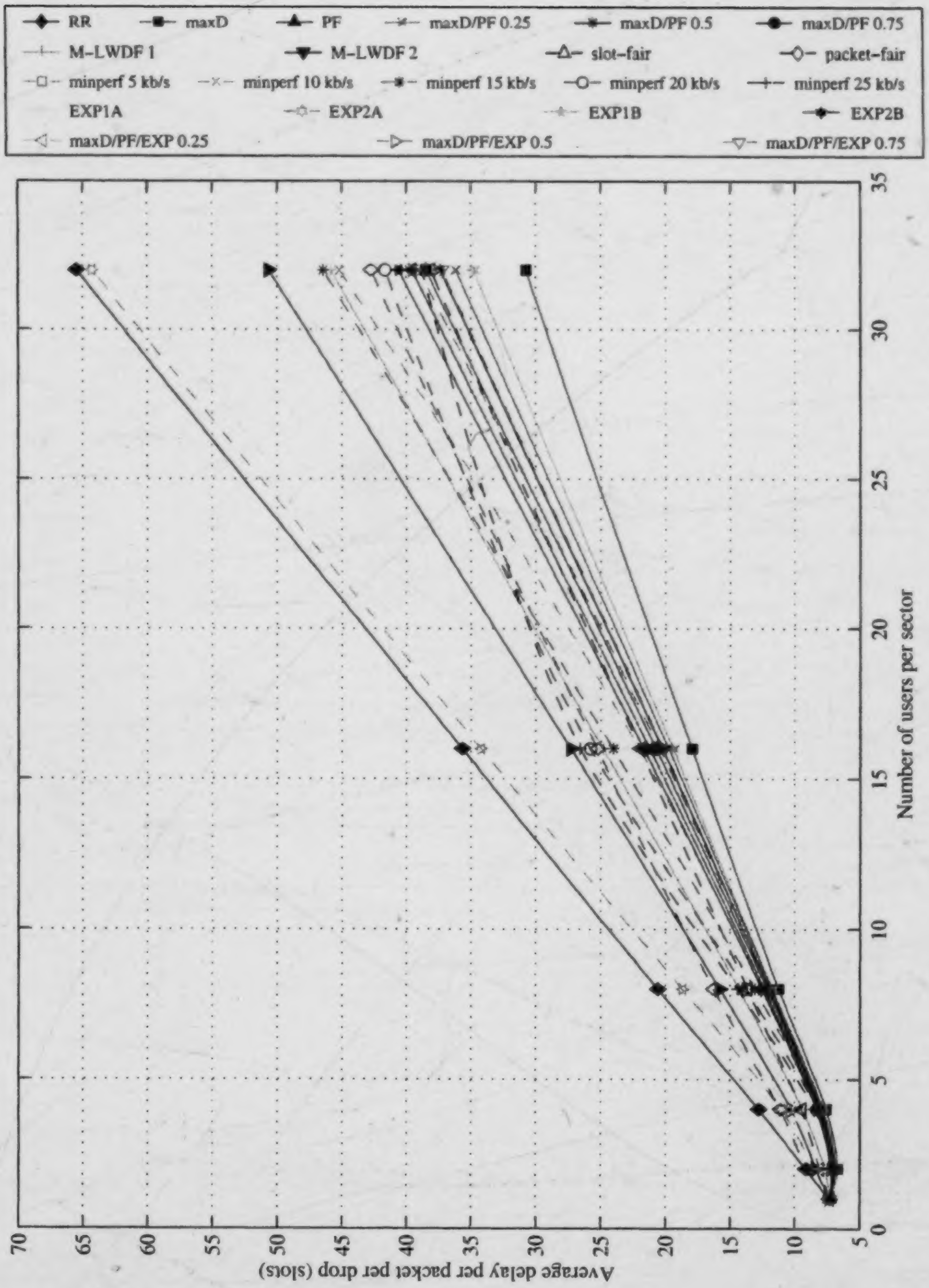


Figure B.10: Average delay per packet per drop versus number of users per sector in the ITU pedestrian B channel

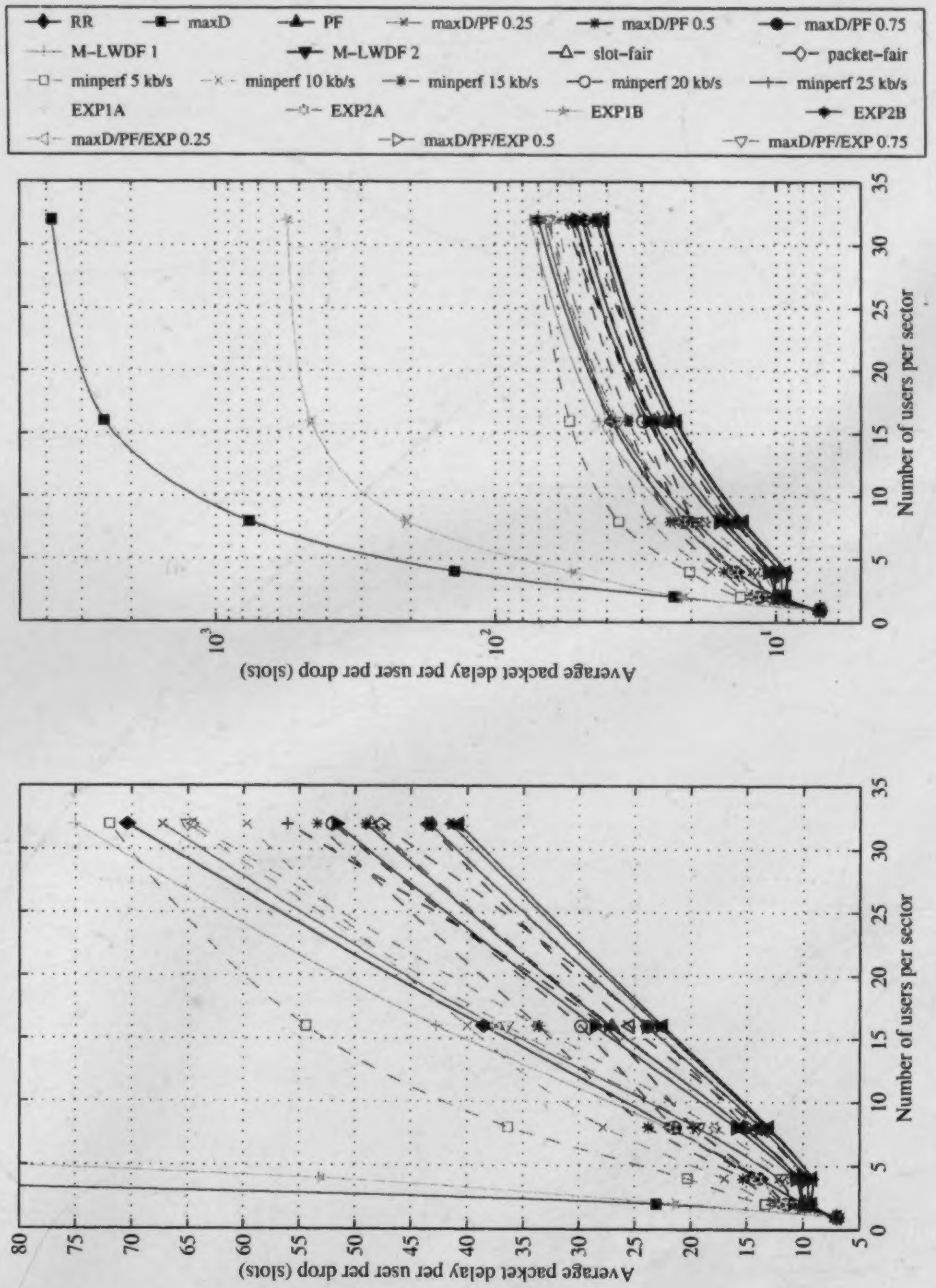


Figure B.11: Average packet delay per user per drop versus number of users per sector in the ITU pedestrian A channel on a linear and a logarithmic scale

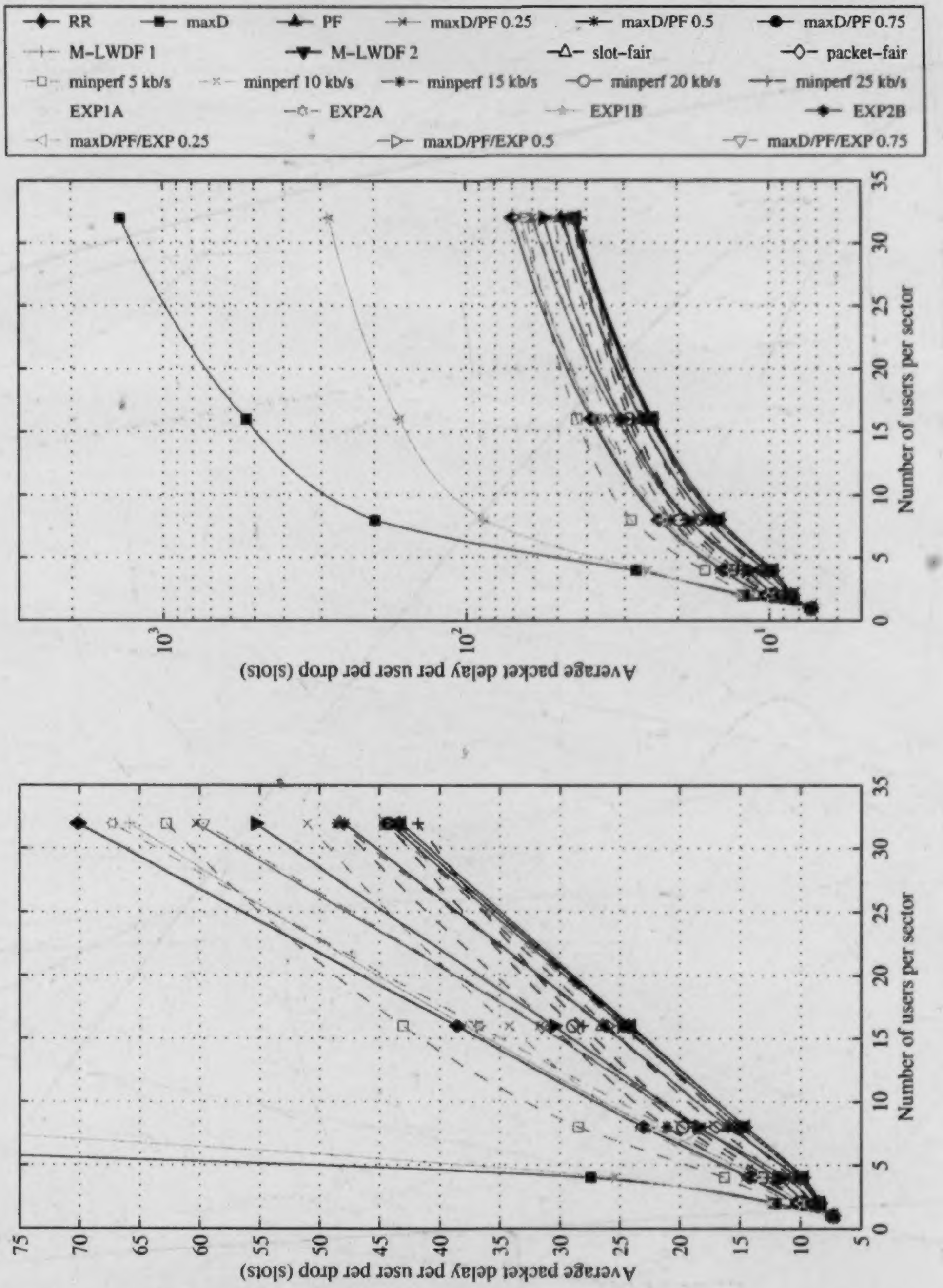


Figure B.12: Average packet delay per user per drop versus number of users per sector in the ITU pedestrian B channel on a linear and a logarithmic scale

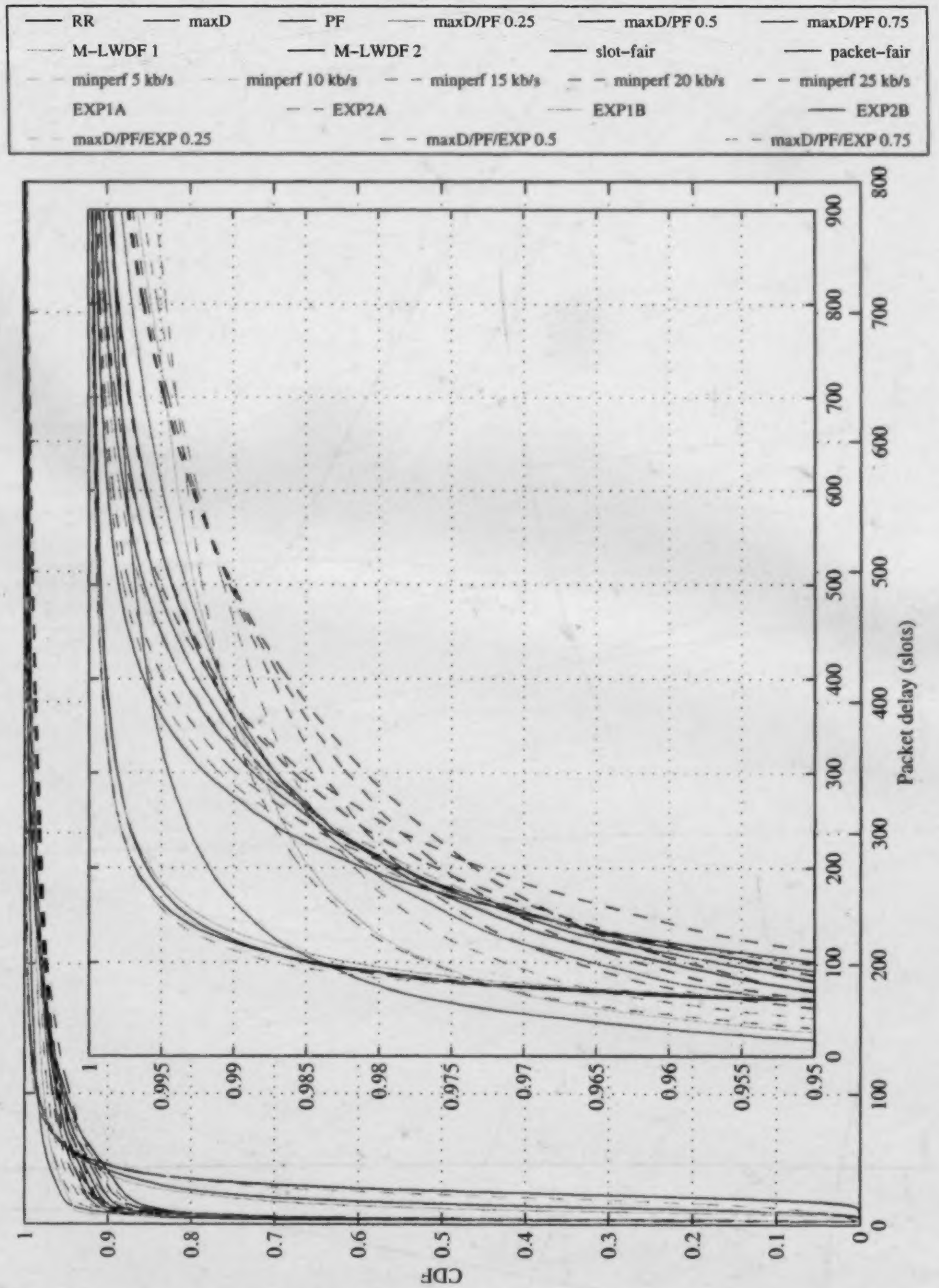


Figure B.13: Cumulative distribution functions of delays experienced by packets in the ITU pedestrian A channel with 16 users per sector

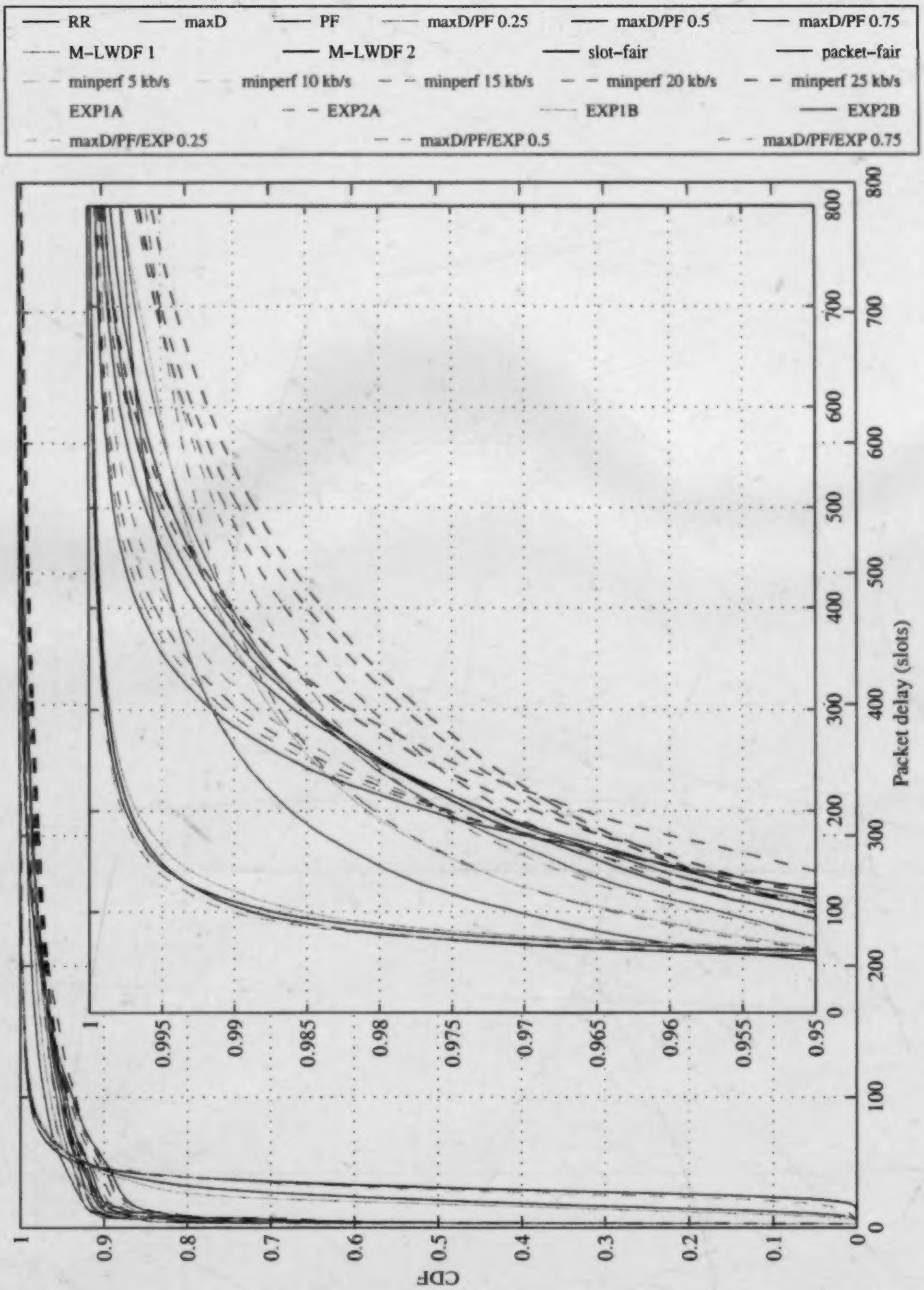


Figure B.14: Cumulative distribution functions of delays experienced by packets in the ITU pedestrian B channel with 16 users per sector

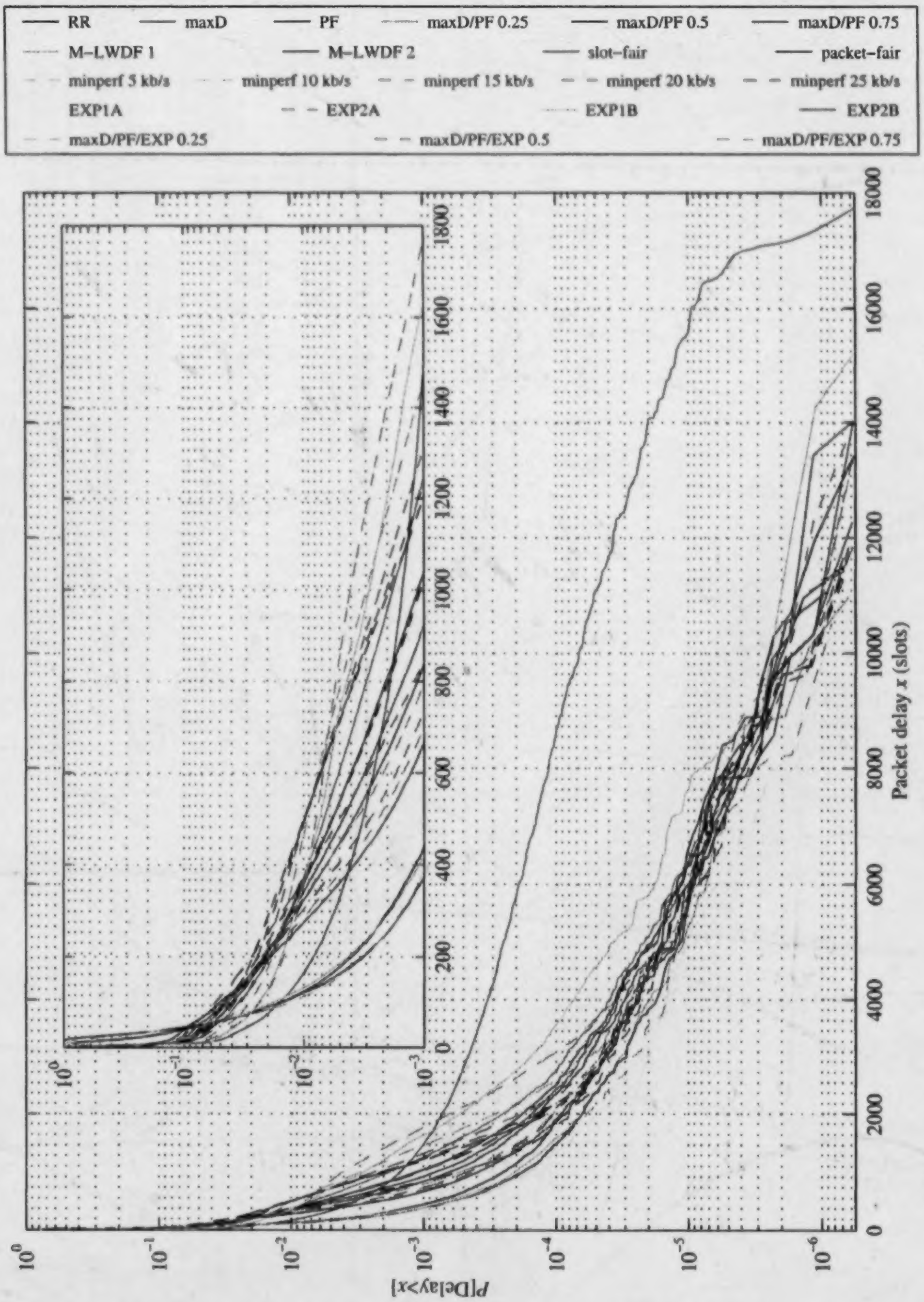


Figure B.15: Probability of delays experienced by packets being larger than a given value in the ITU pedestrian A channel with 16 users per sector

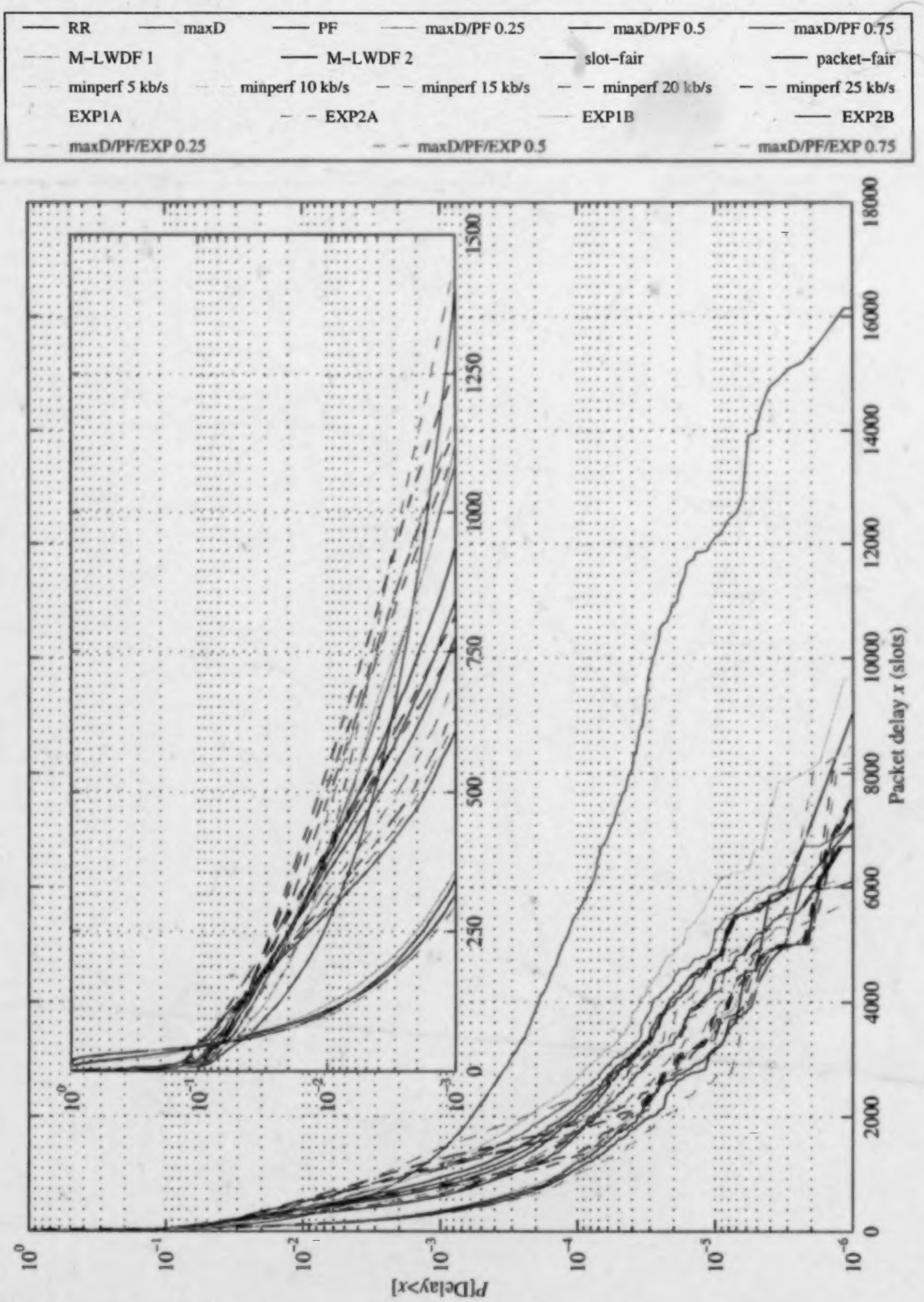


Figure B.16: Probability of delays experienced by packets being larger than a given value in the ITU pedestrian B channel with 16 users per sector

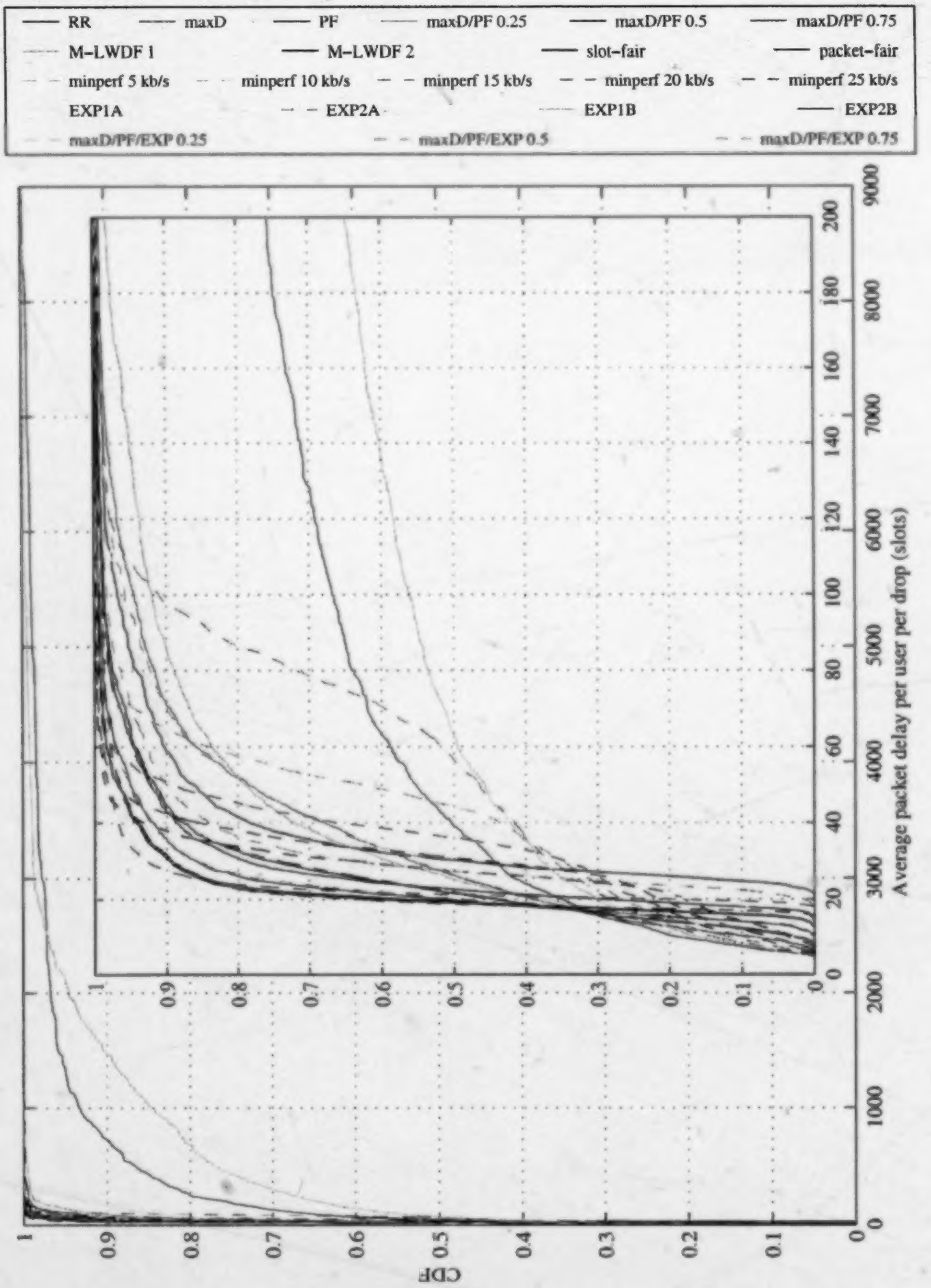


Figure B.17: Distributions of average packet delay per user for users who receive at least one packet in the ITU pedestrian A channel with 16 users per sector

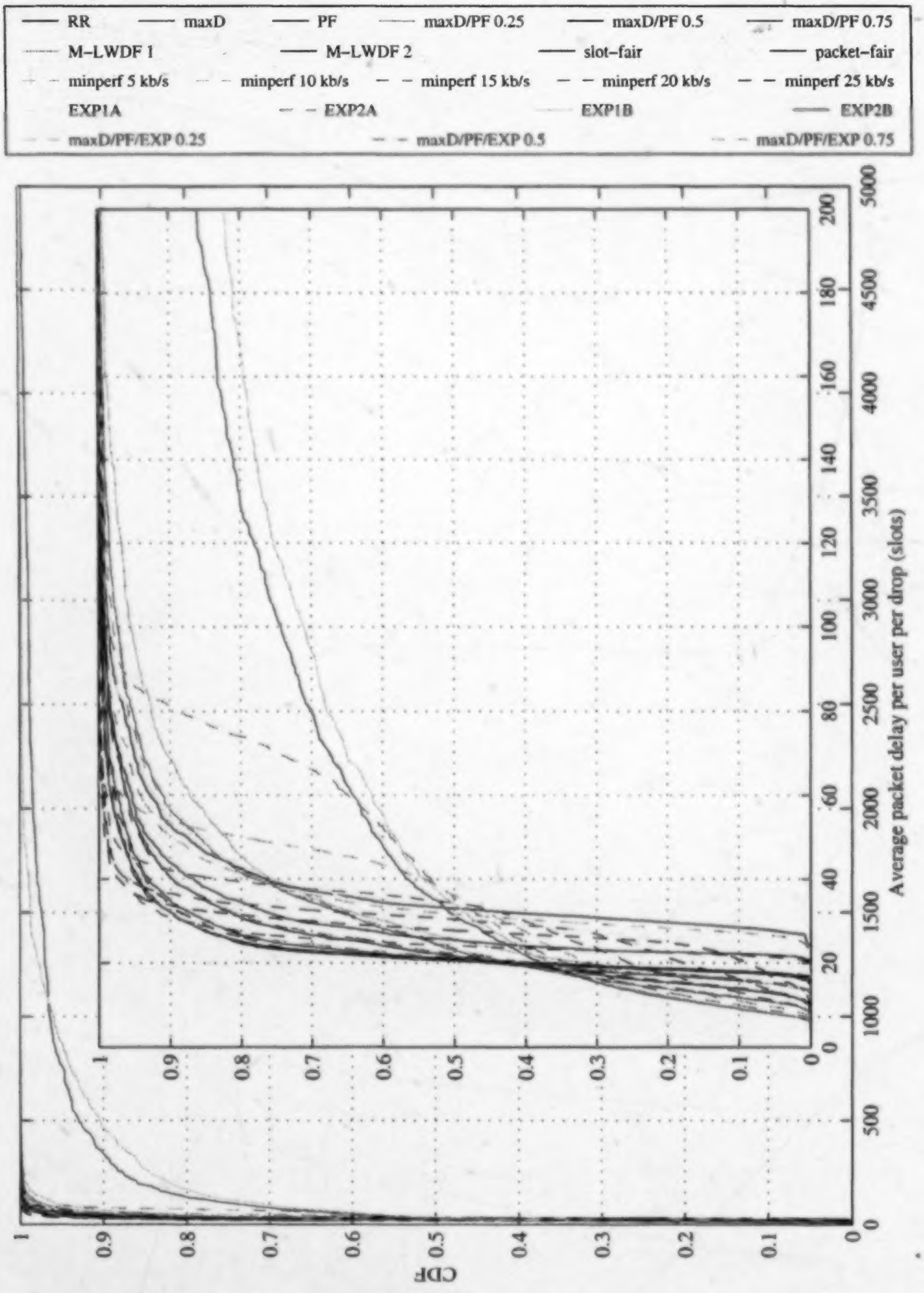


Figure B.18: Distributions of average packet delay per user for users who receive at least on packet in the ITU pedestrian B channel with 16 users per sector

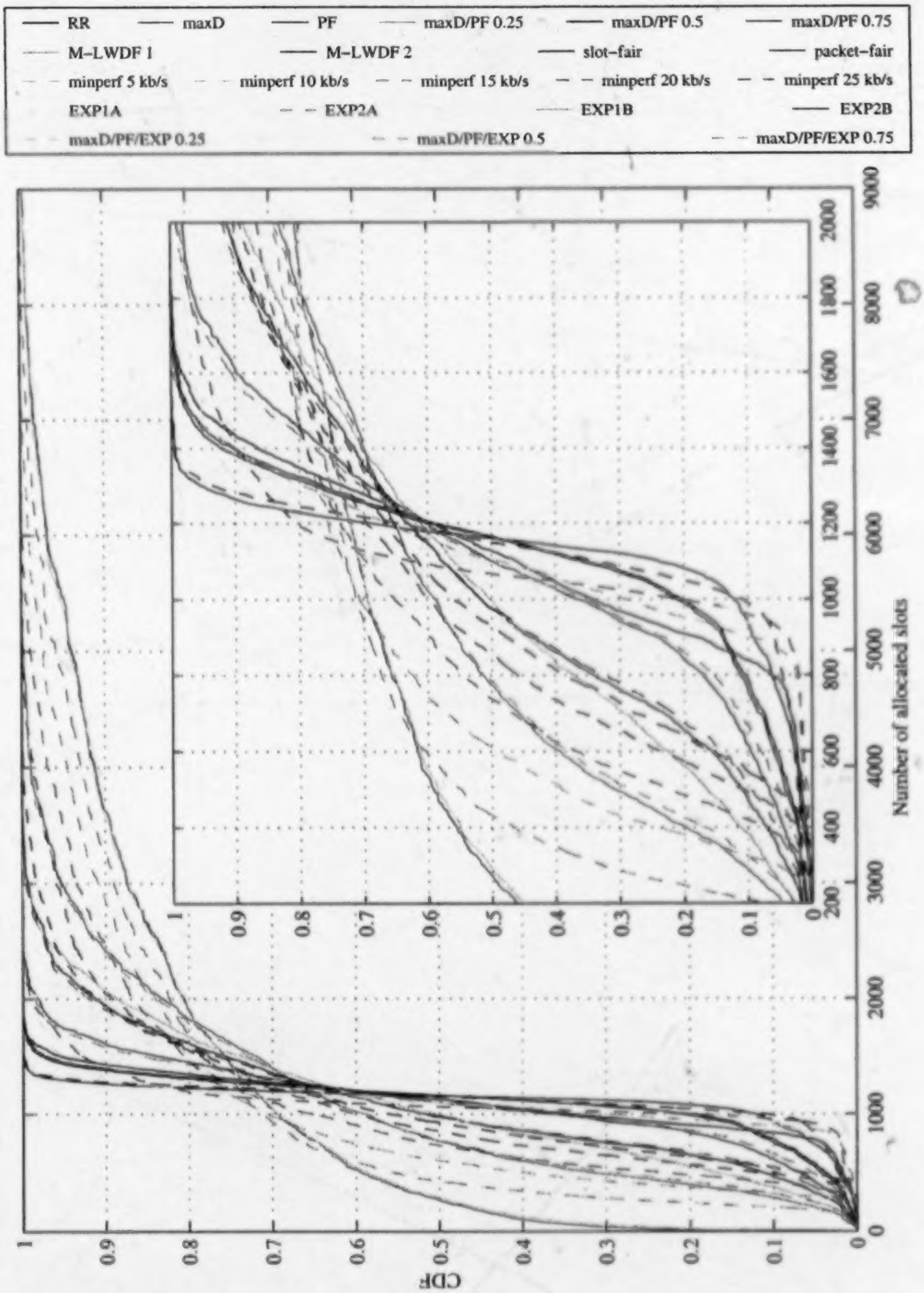


Figure B.19: Distributions of allocated slots per user per drop in the ITU pedestrian A channel with 16 users per sector

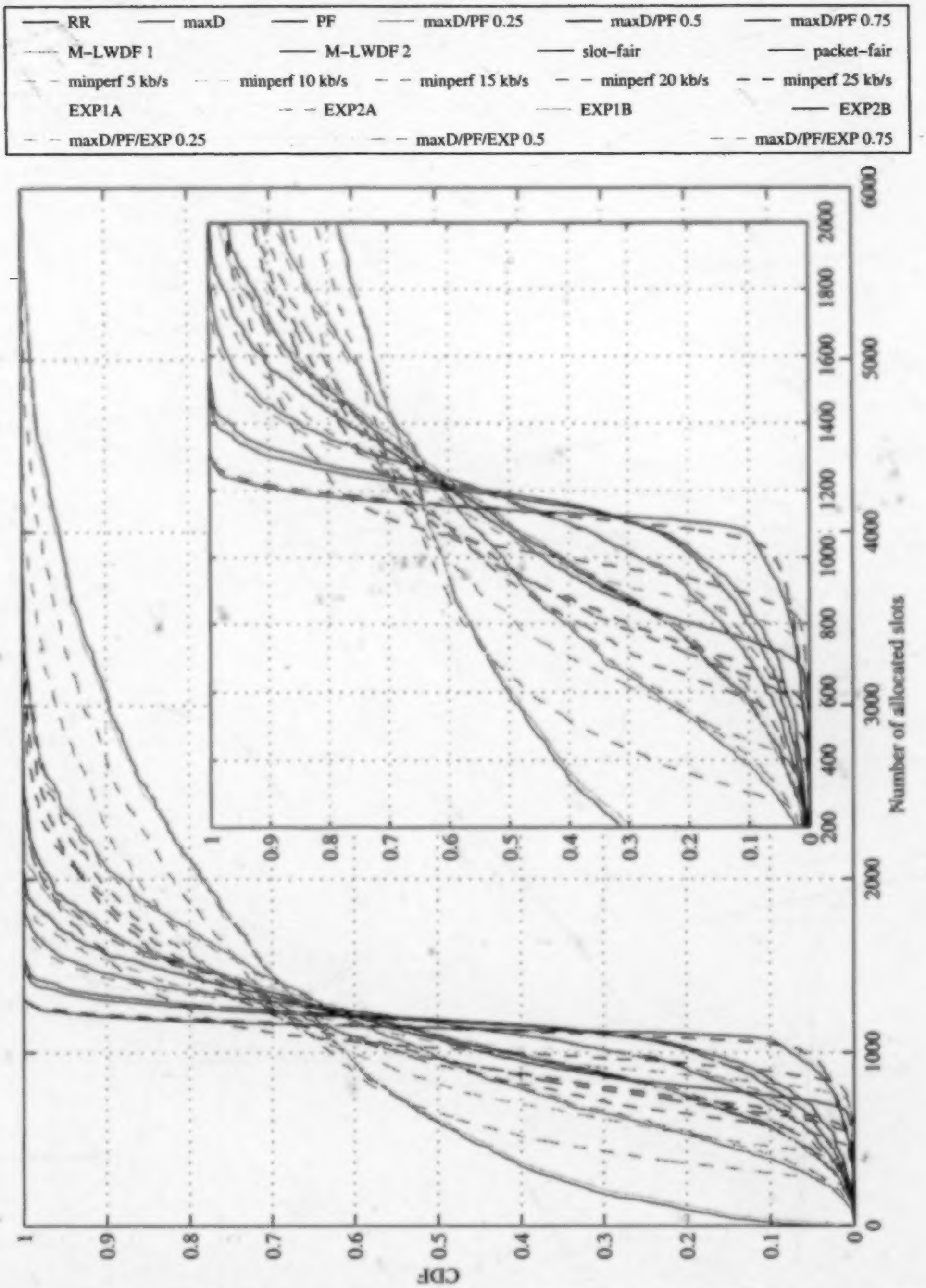


Figure B.20: Distributions of allocated slots per user per drop in the ITU pedestrian B channel with 16 users per sector

Appendix C: Pedestrian Channel Results using Perfect Multislot Prediction

This appendix contains graphs showing the simulation results for the ITU pedestrian A and B channels when perfect multislot prediction was used to determine the transmission format each user could support.

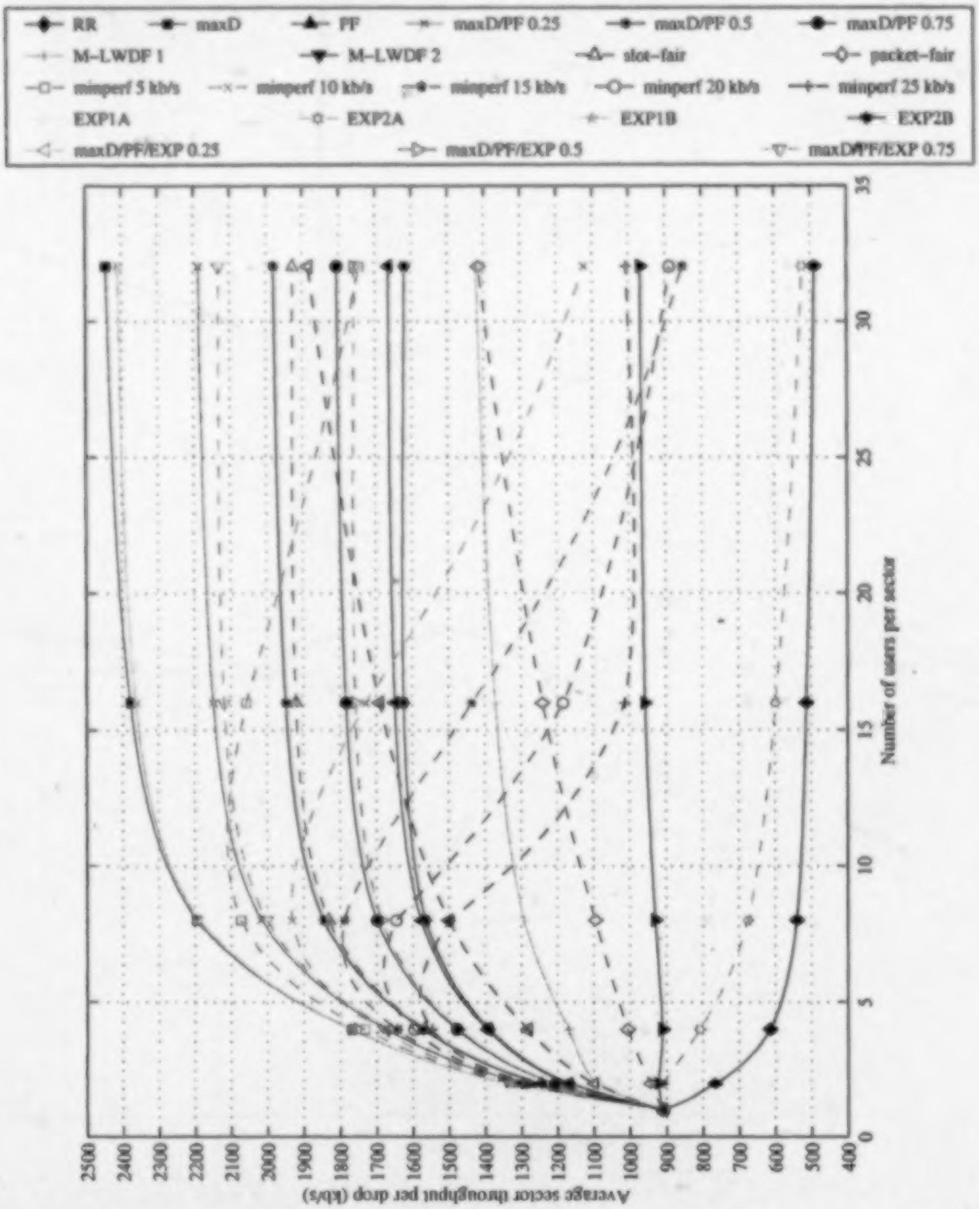


Figure C.1: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian A channel using perfect multislot prediction

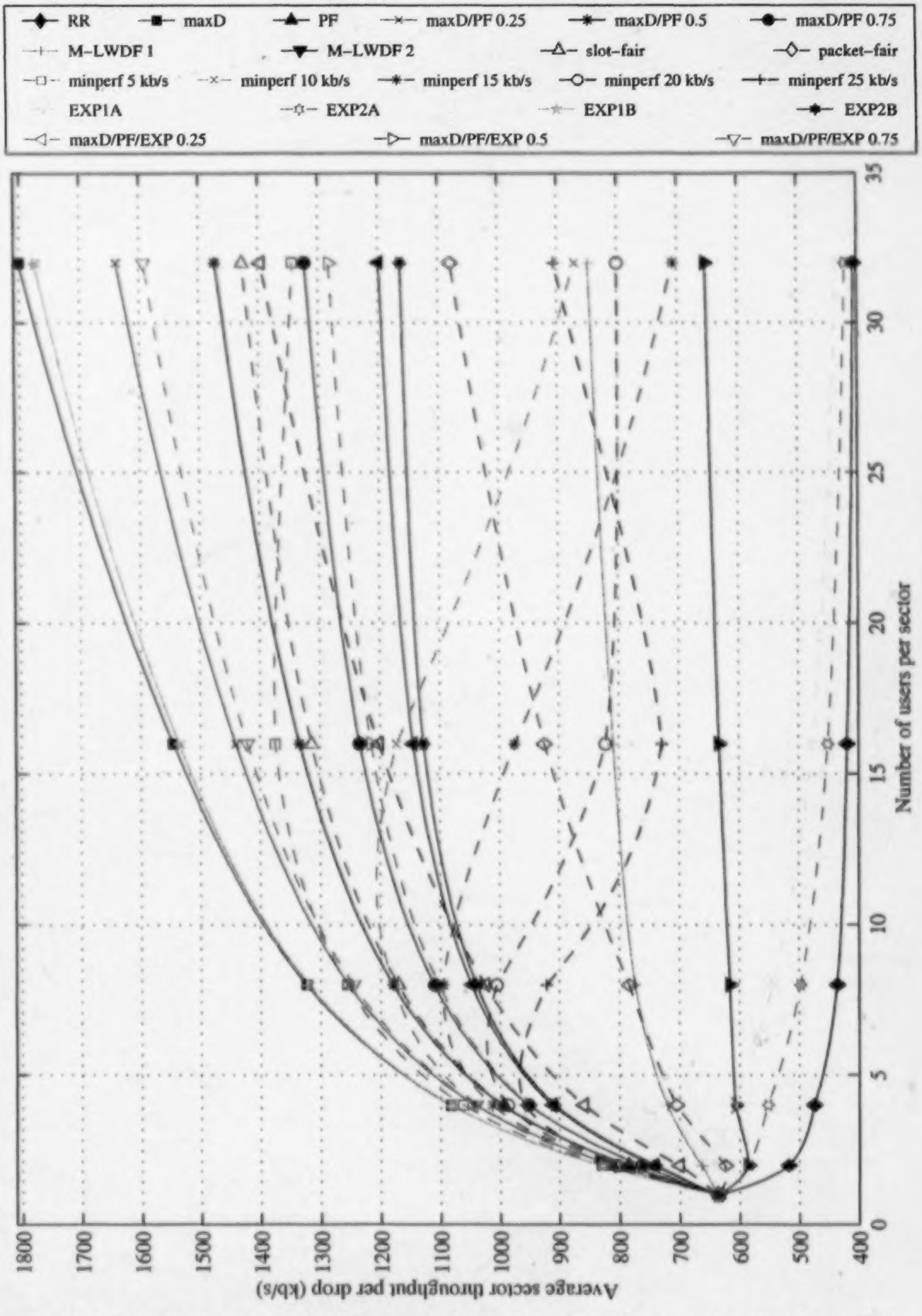


Figure C.2: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian B channel using perfect multislot prediction

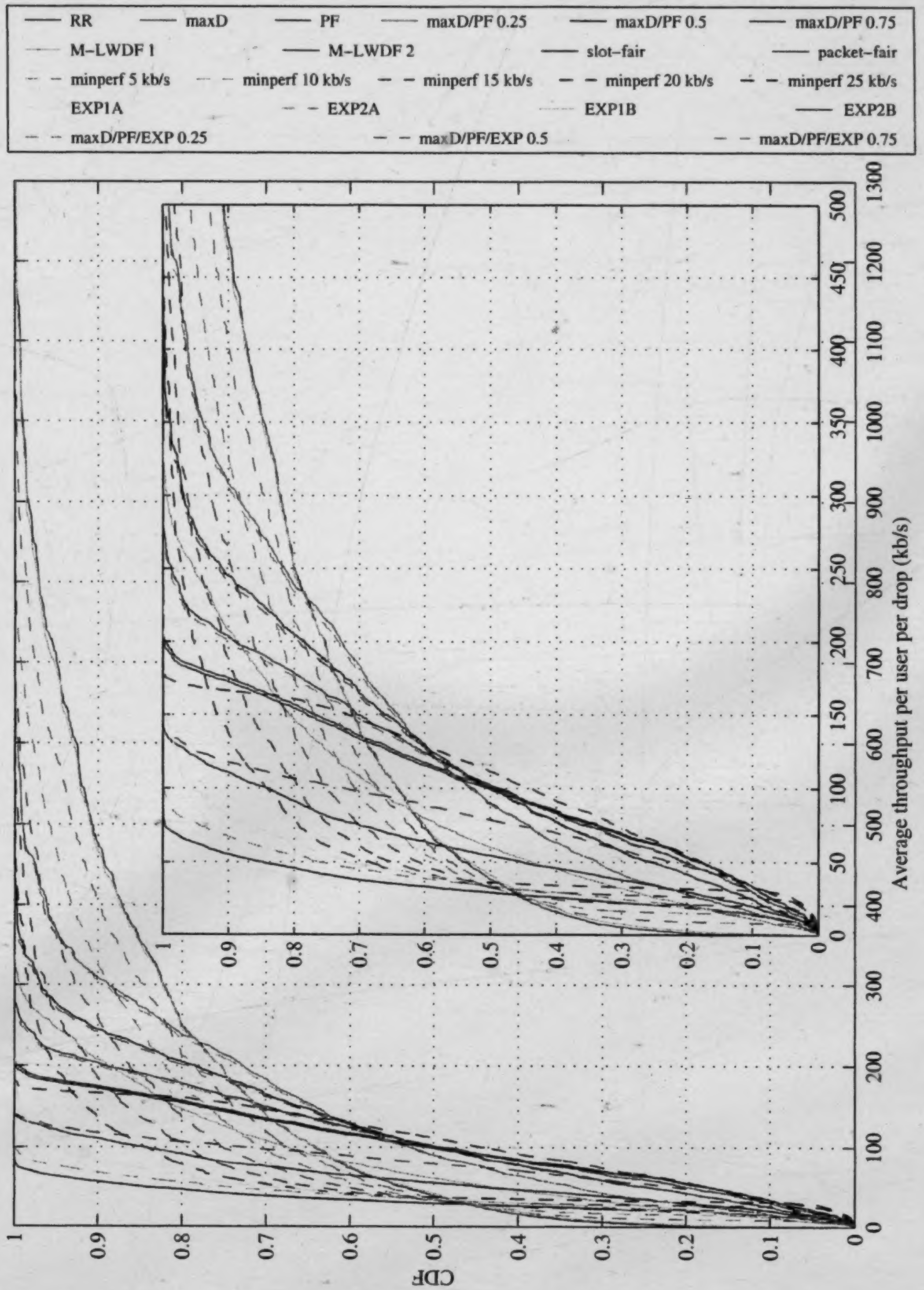


Figure C.3: Distributions of average throughput per user per drop in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction

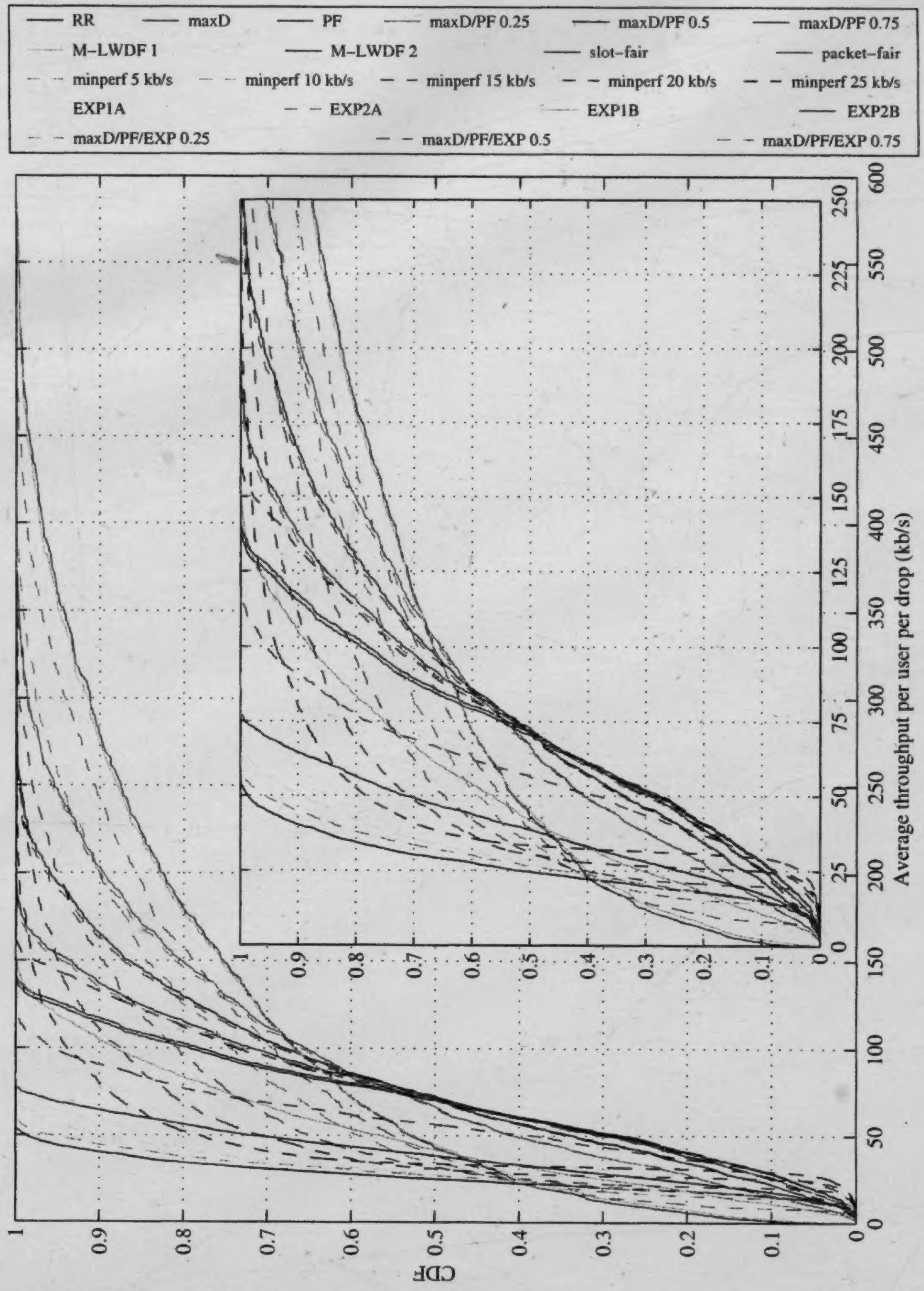


Figure C.4: Distributions of average throughput per user per drop in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction

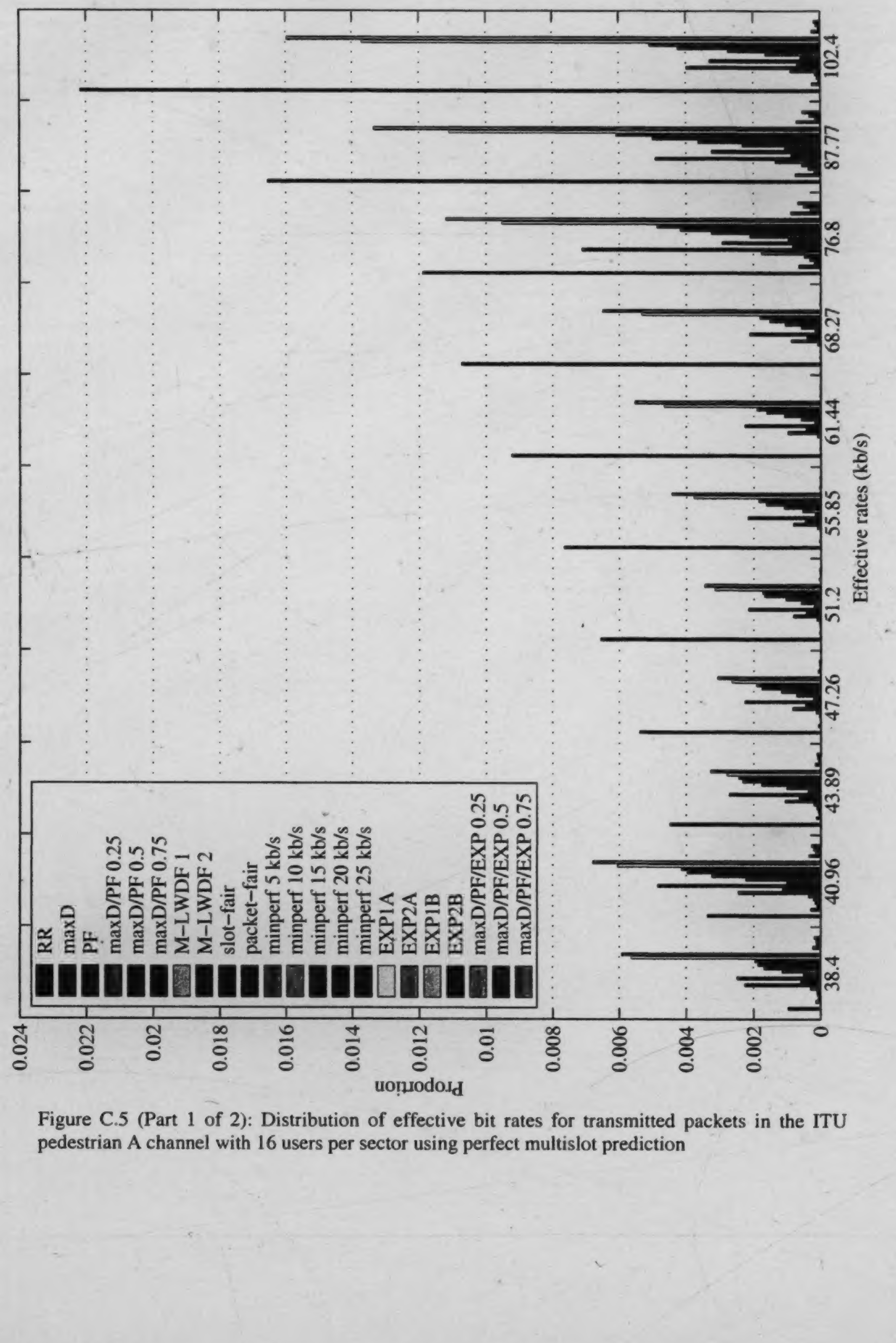


Figure C.5 (Part 1 of 2): Distribution of effective bit rates for transmitted packets in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction

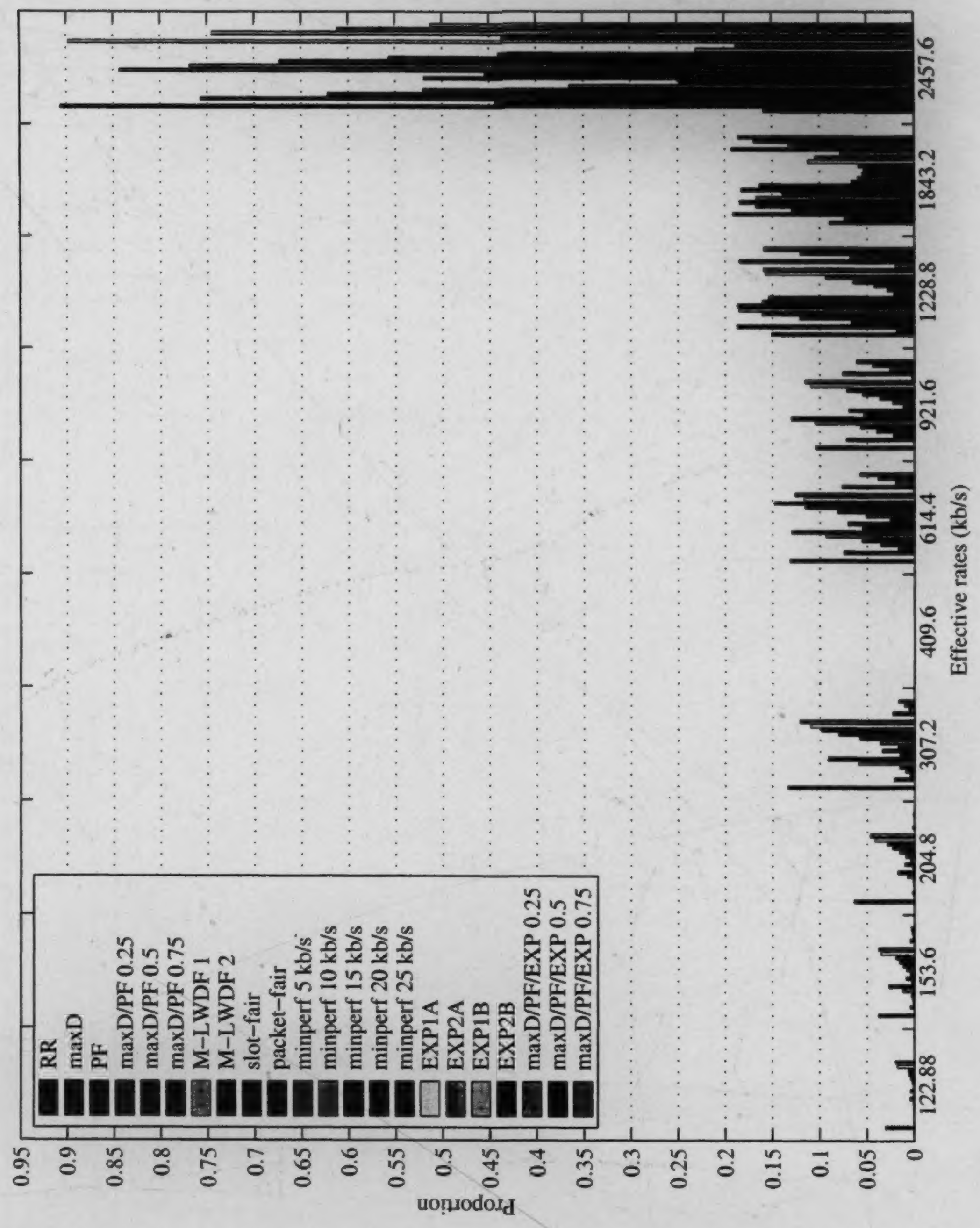


Figure C.5 (Part 2 of 2): Distribution of effective bit rates for transmitted packets in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction

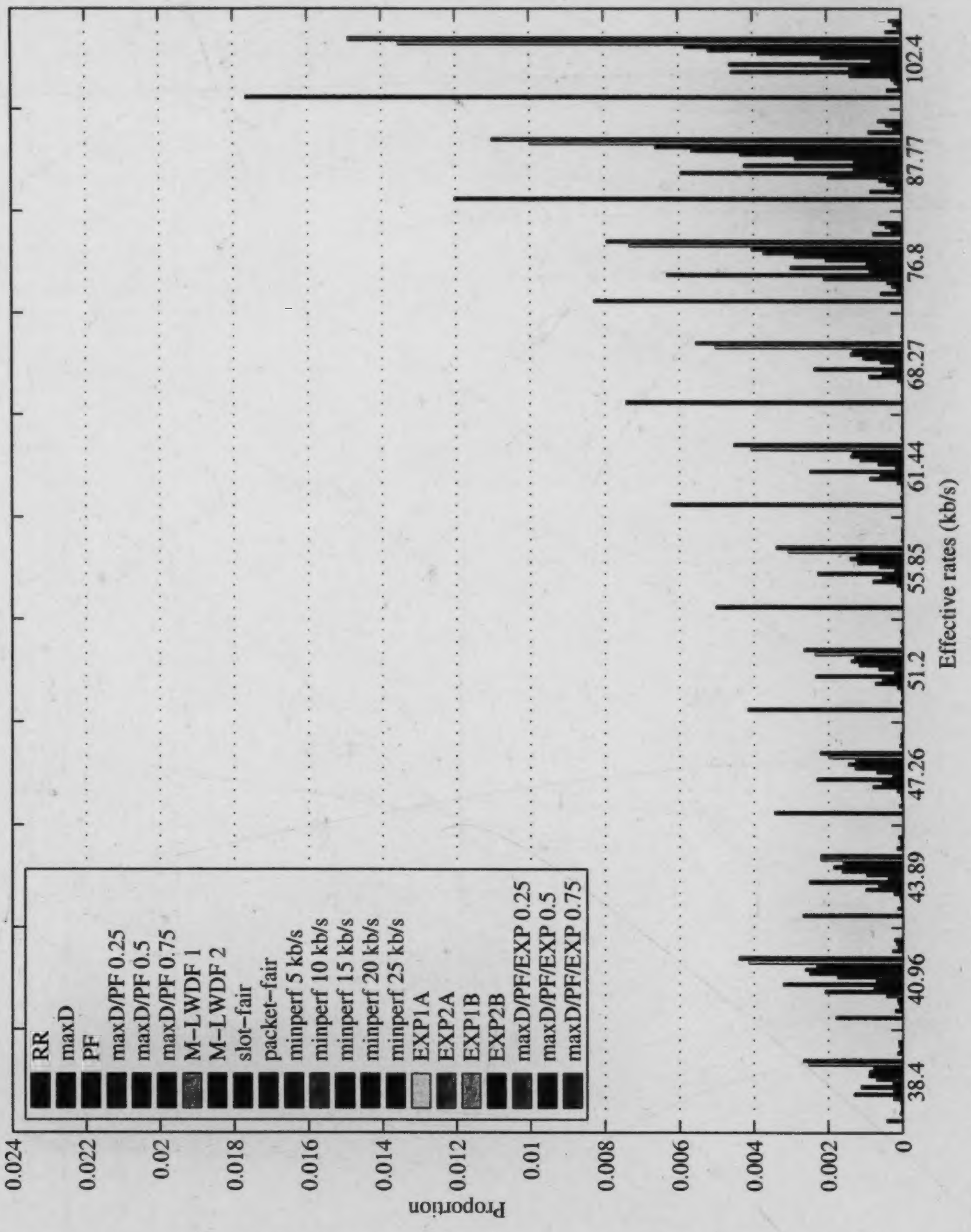


Figure C.6 (Part 1 of 2): Distribution of effective bit rates for transmitted packets in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction

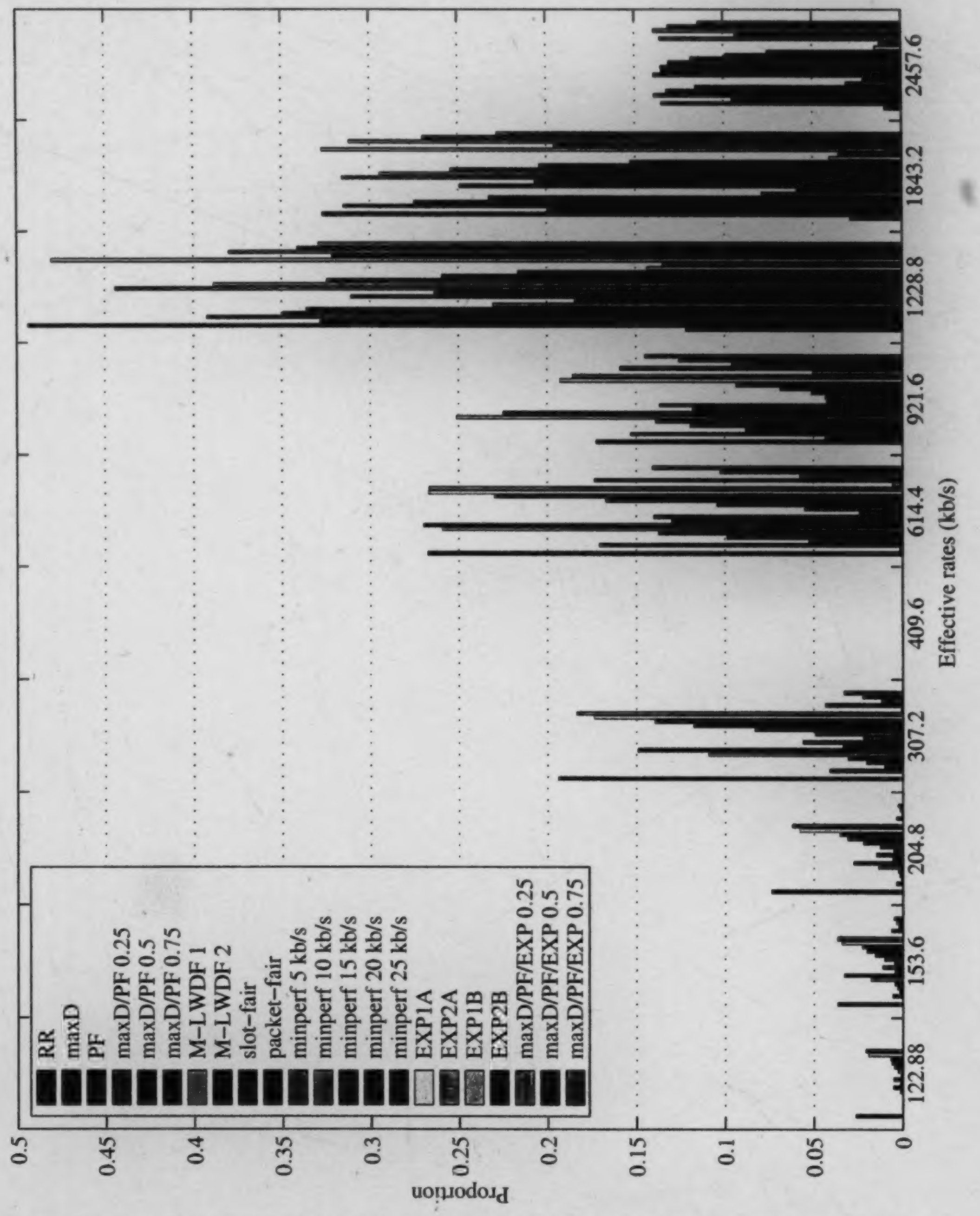
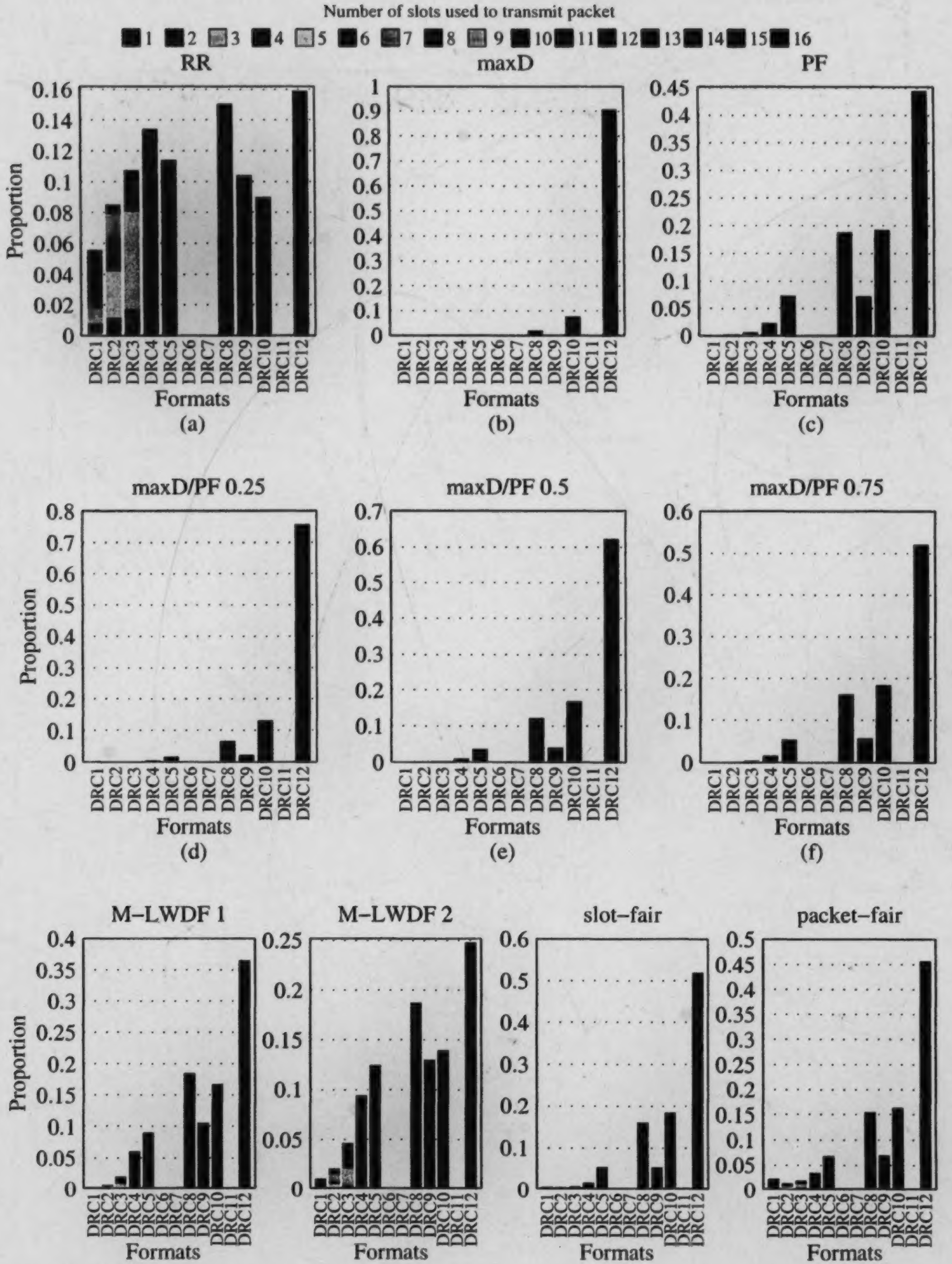


Figure C.6 (Part 2 of 2): Distribution of effective bit rates for transmitted packets in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction



(a) (b) (c)

(d) (e) (f)

(g) (h) (i) (j)

Figure C.7 (Part 1 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction

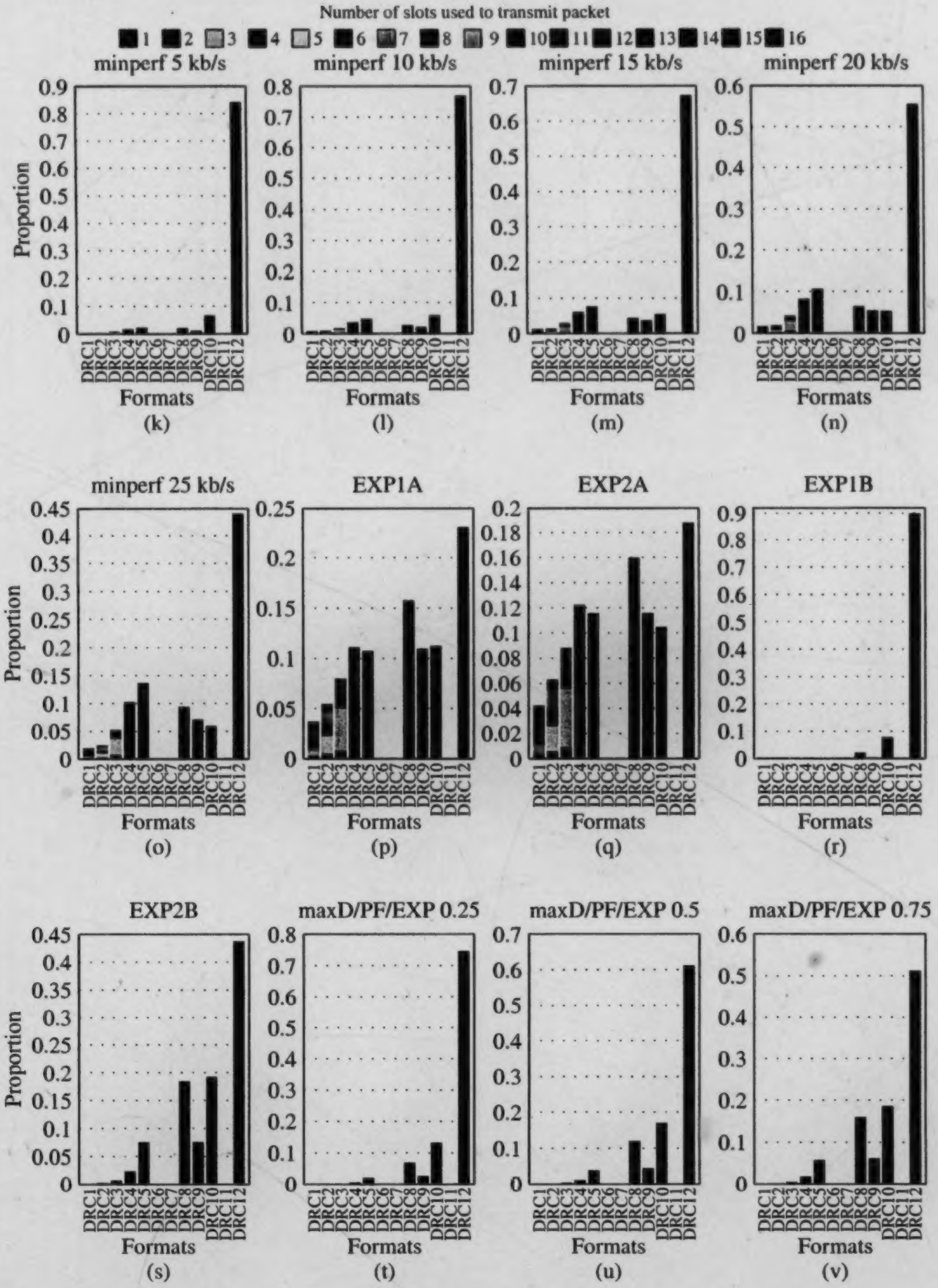


Figure C.7 (Part 2 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction

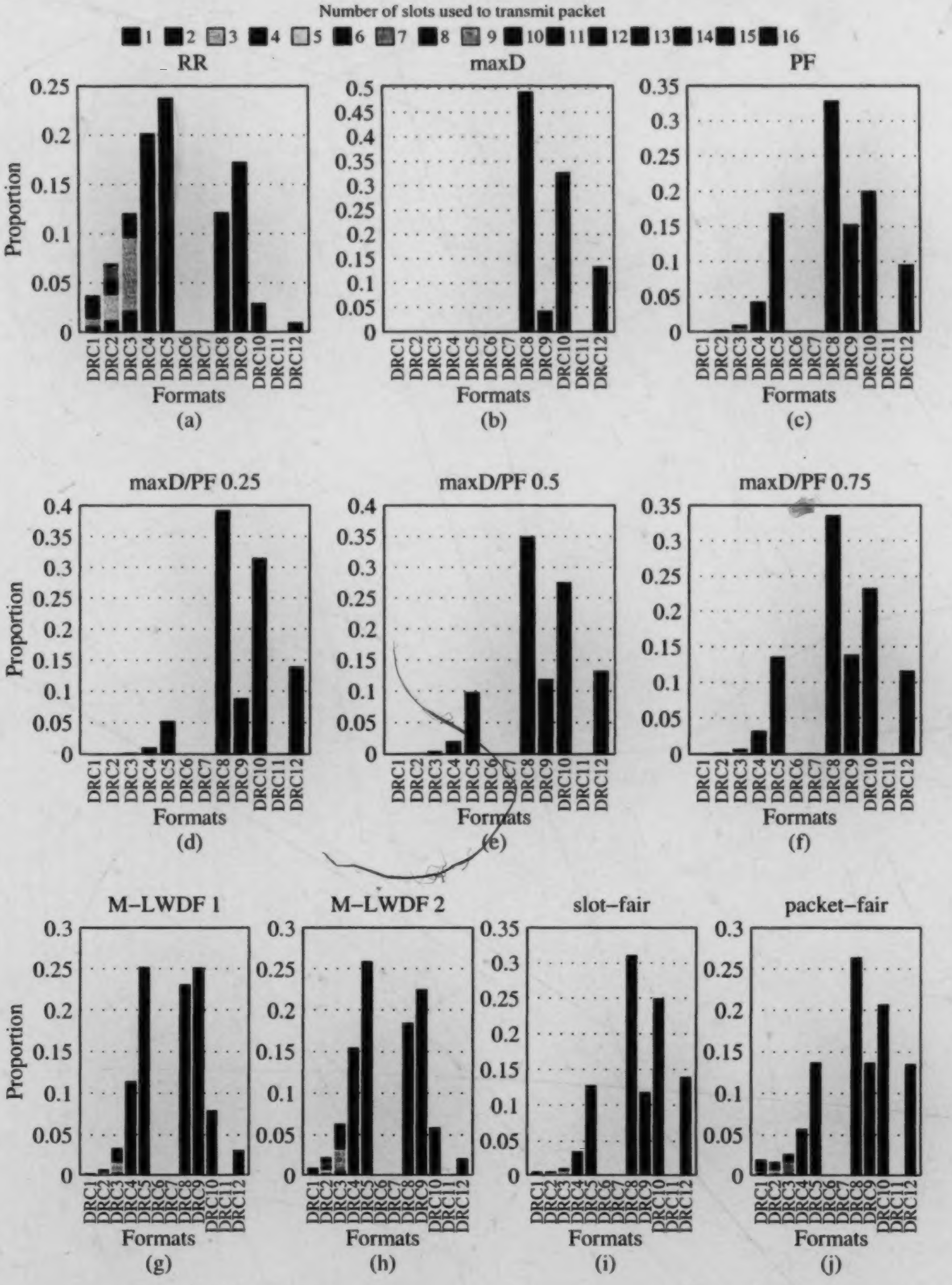


Figure C.8 (Part 1 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction

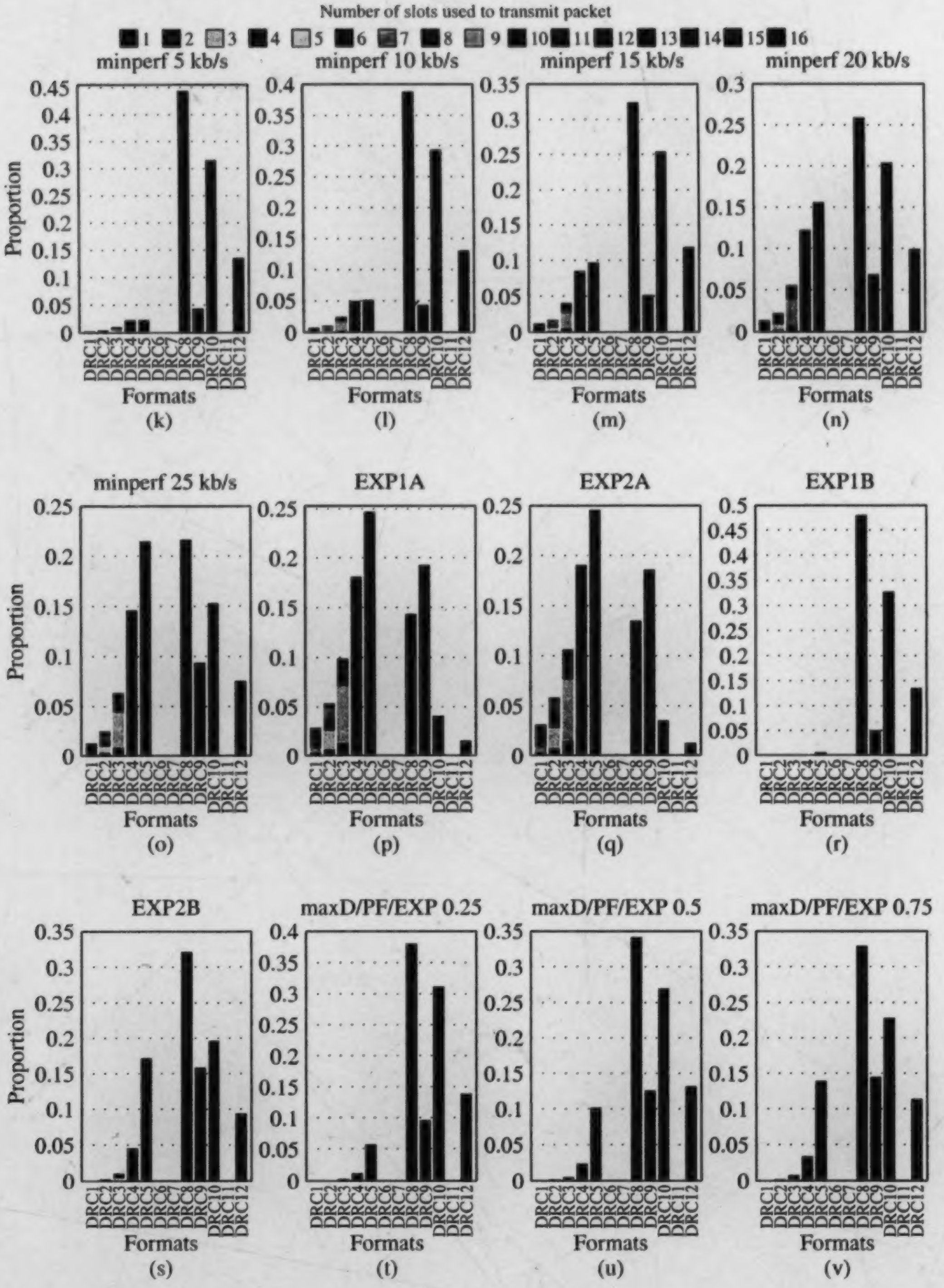


Figure C.8 (Part 2 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction

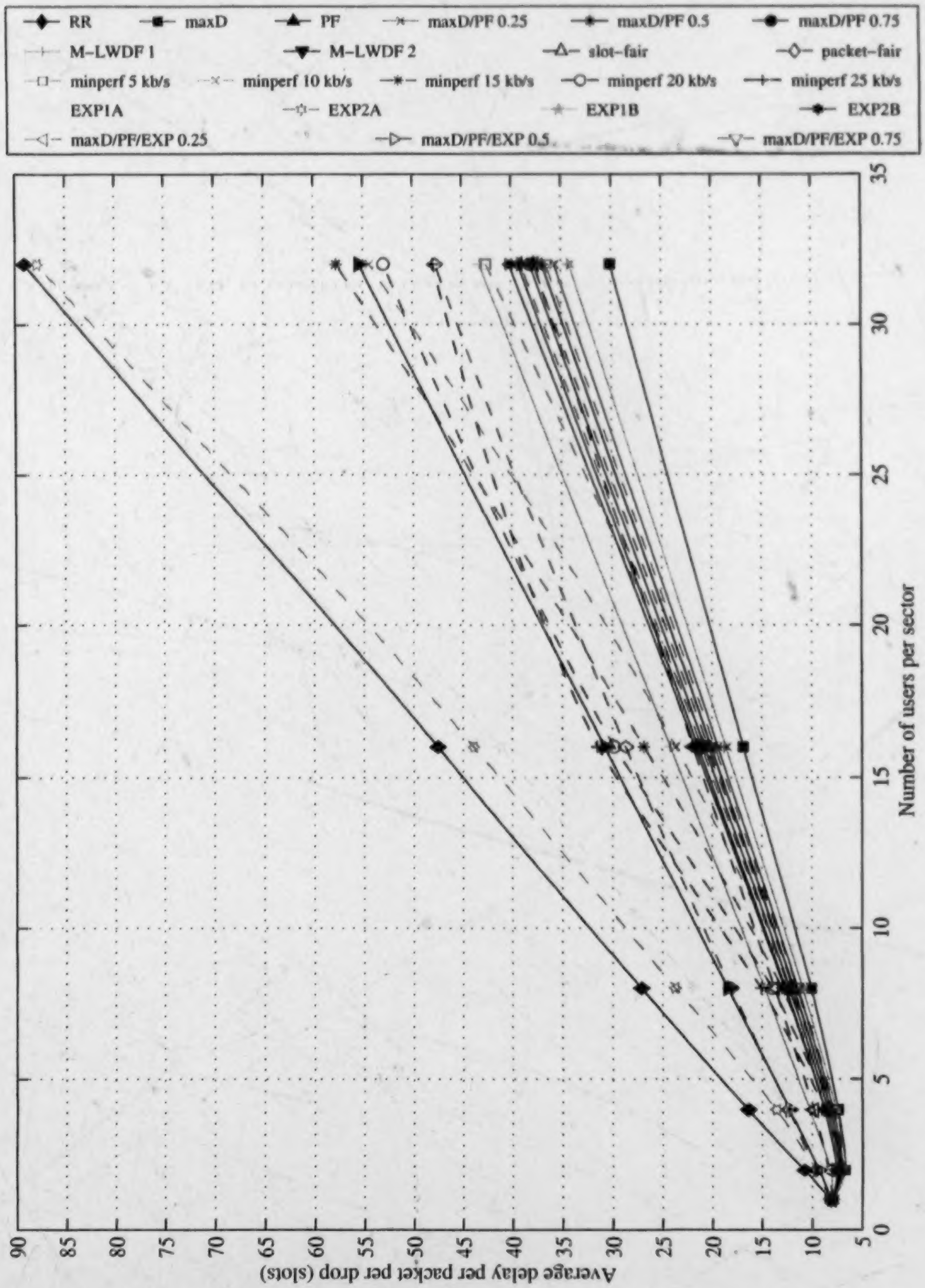


Figure C.9: Average delay per packet per drop versus number of users per sector in the ITU pedestrian A channel using perfect multislot prediction

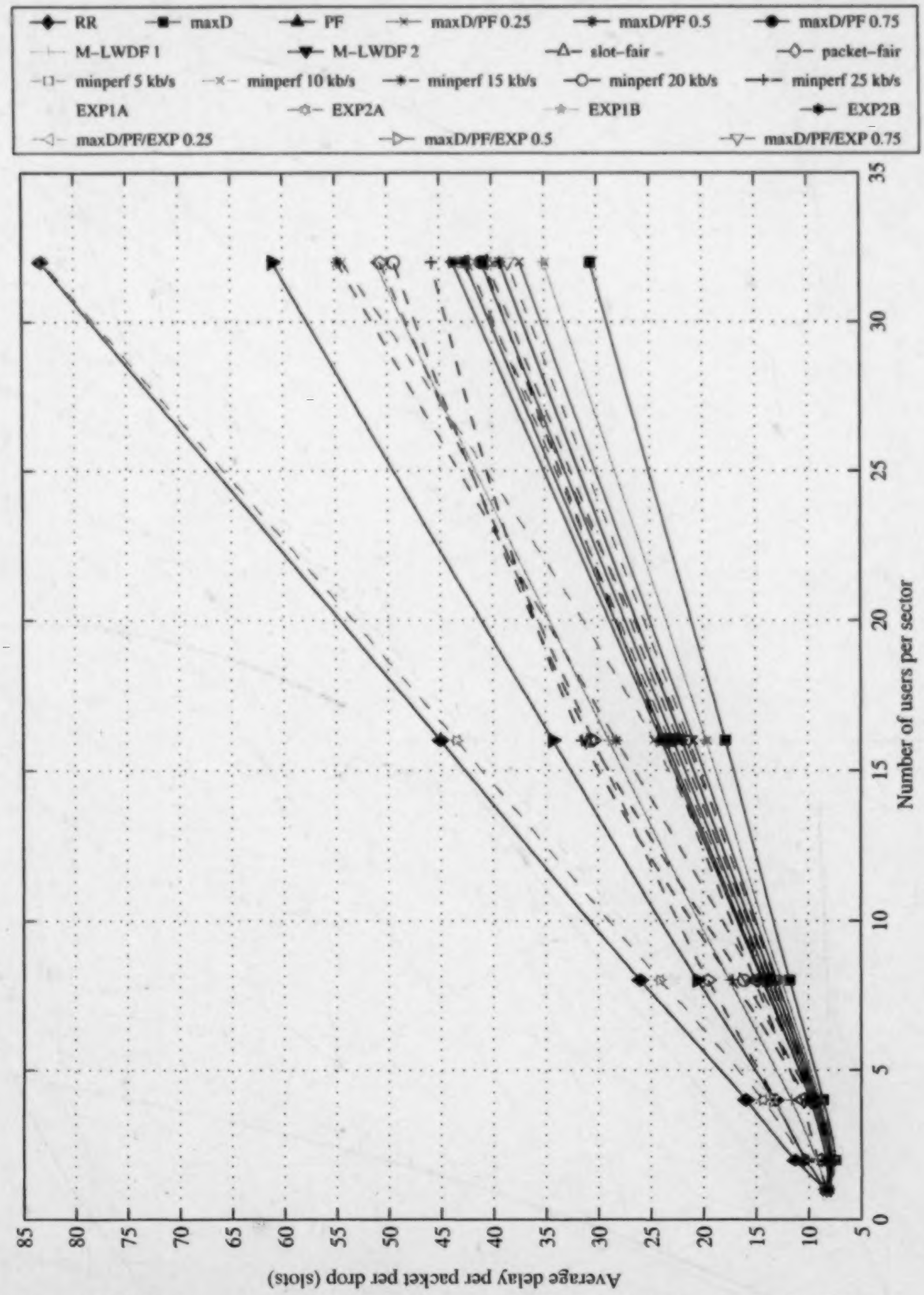


Figure C.10: Average delay per packet per drop versus number of users per sector in the ITU pedestrian B channel using perfect multislot prediction

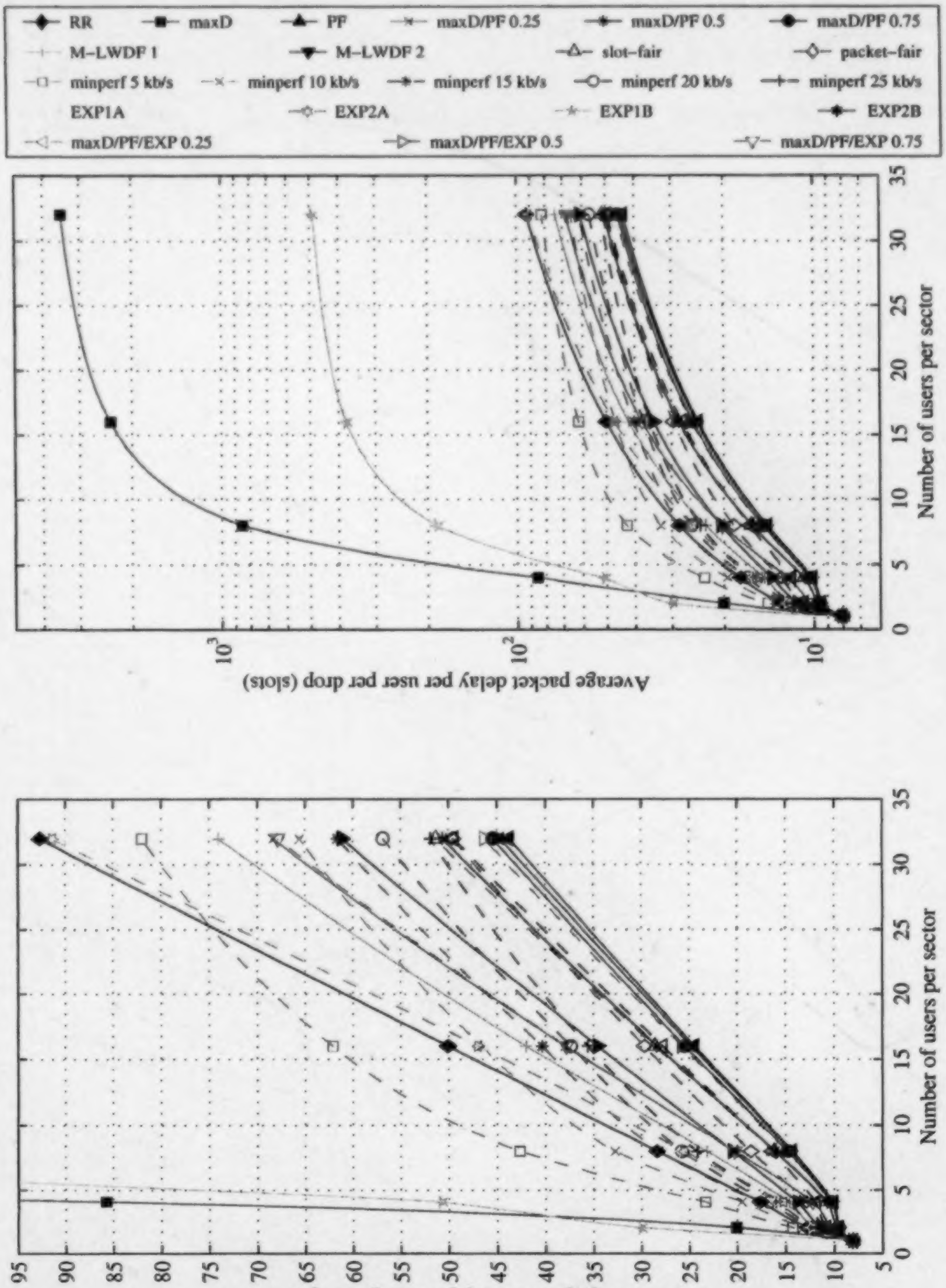


Figure C.11: Average packet delay per user per drop versus number of users per sector in the ITU pedestrian A channel on a linear and a logarithmic scale using perfect multislot prediction

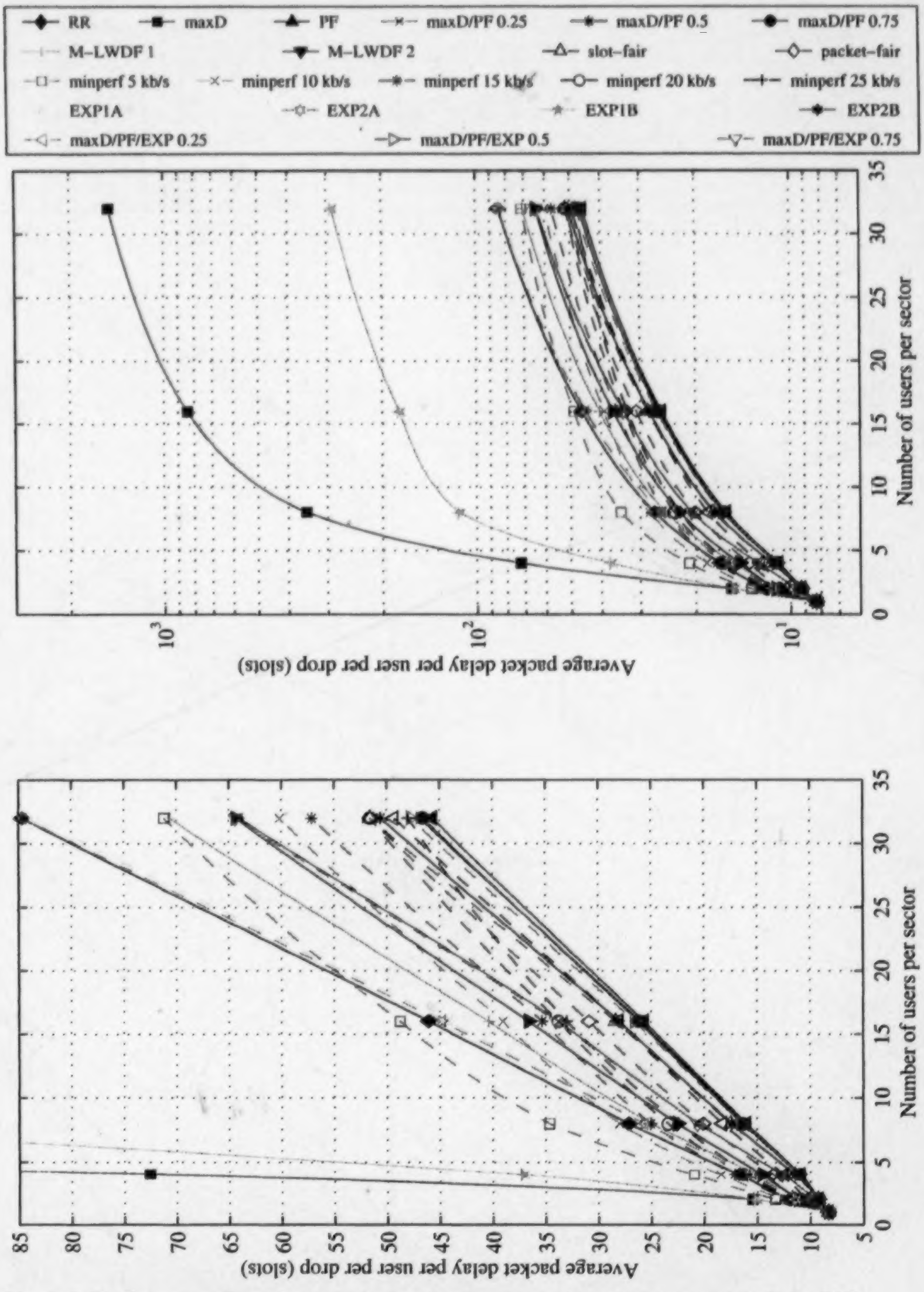


Figure C.12: Average packet delay per user per drop versus number of users per sector in the ITU pedestrian B channel on a linear and a logarithmic scale using perfect multislot prediction

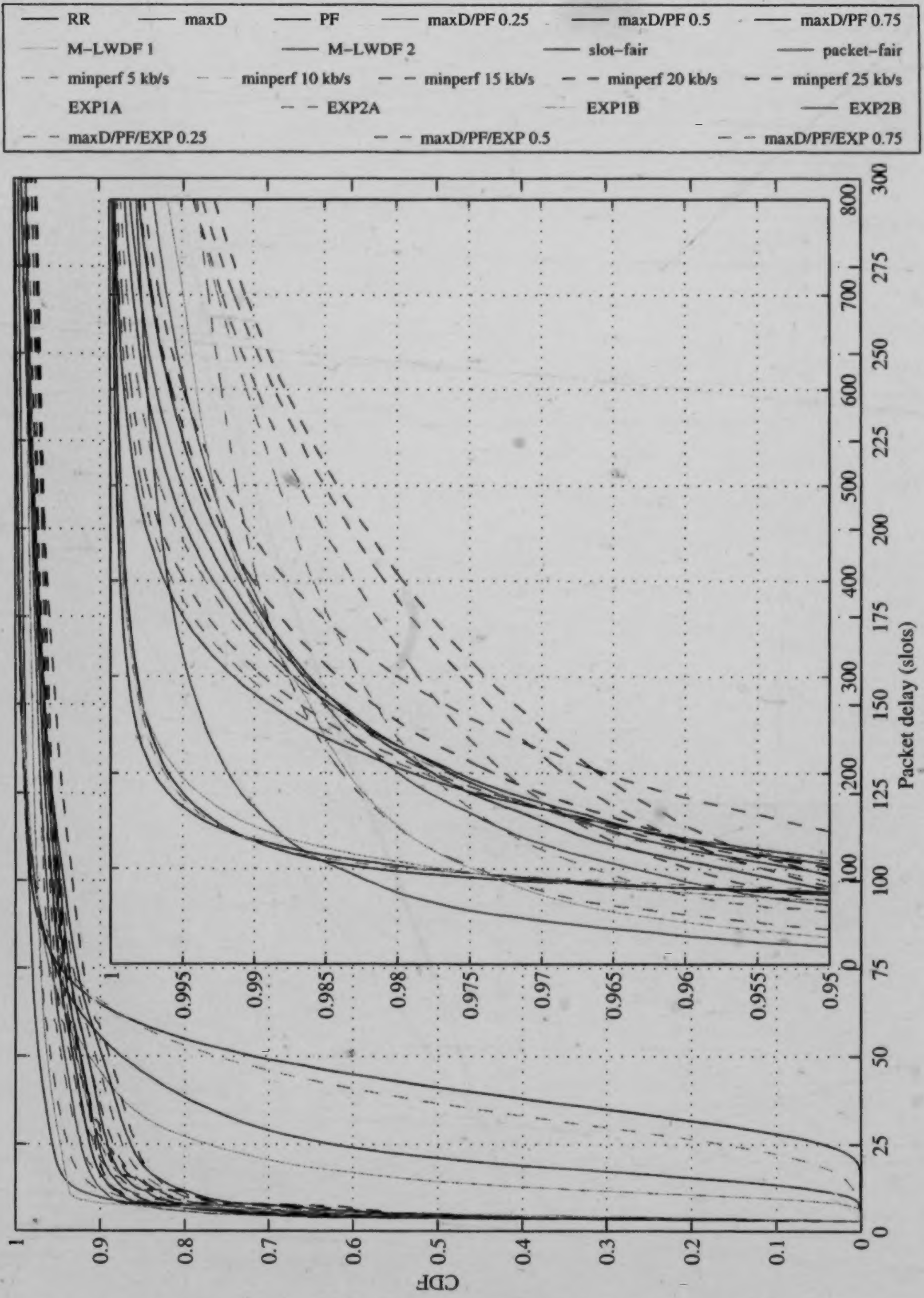


Figure C.13: Cumulative distribution functions of delays experienced by packets in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction

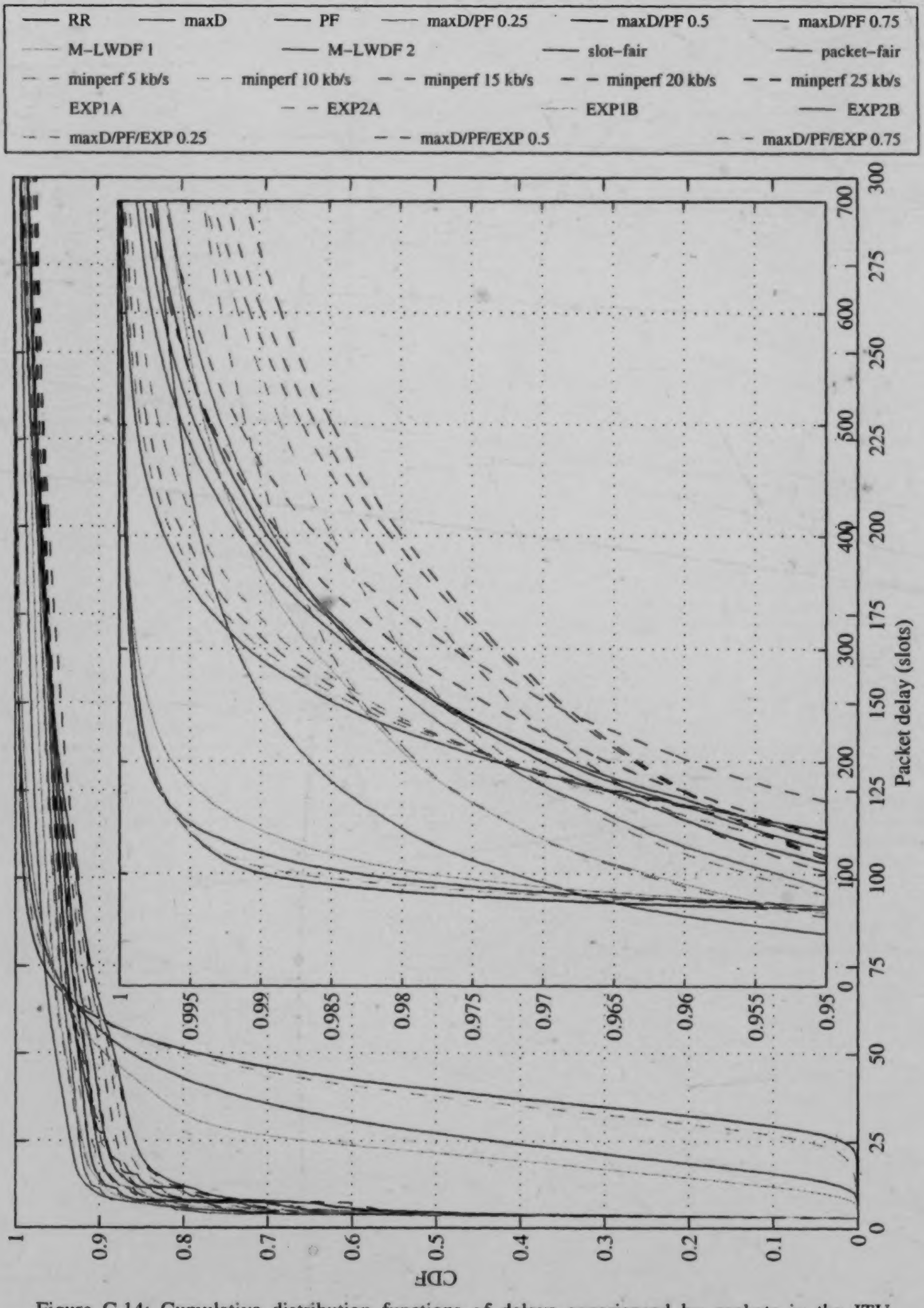


Figure C.14: Cumulative distribution functions of delays experienced by packets in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction

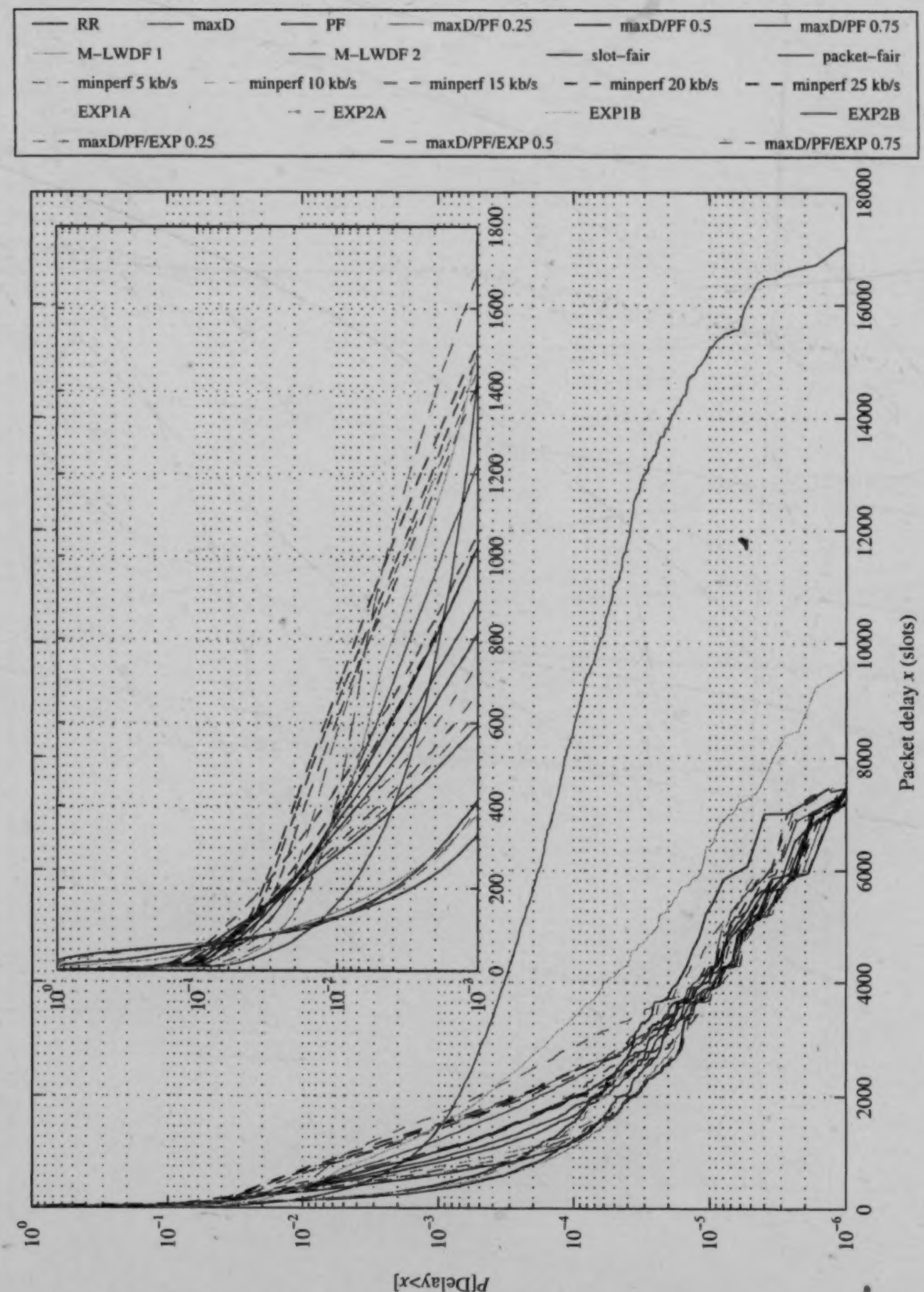


Figure C.15: Probability of delays experienced by packets being larger than a given value in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction

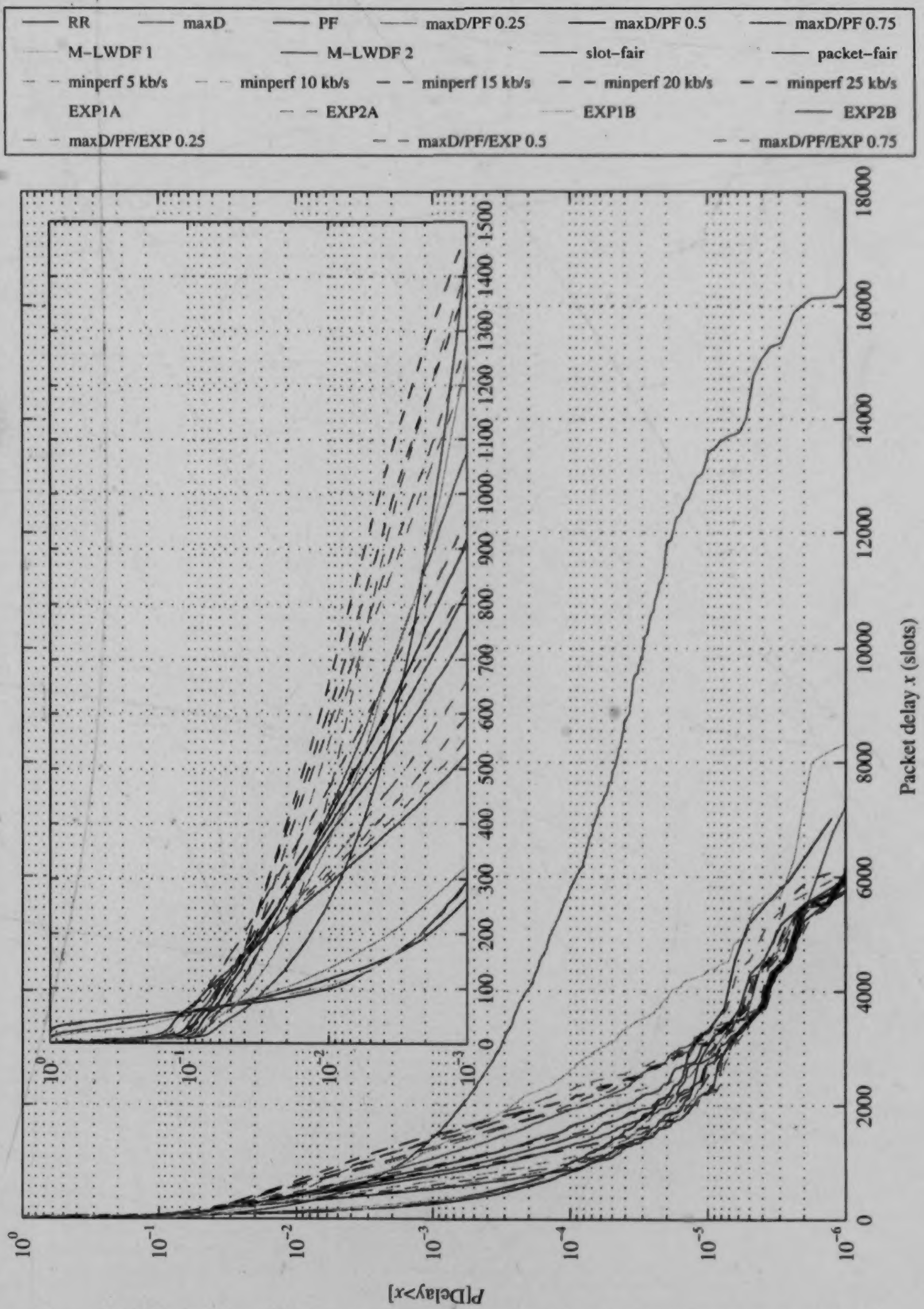


Figure C.16: Probability of delays experienced by packets being larger than a given value in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction

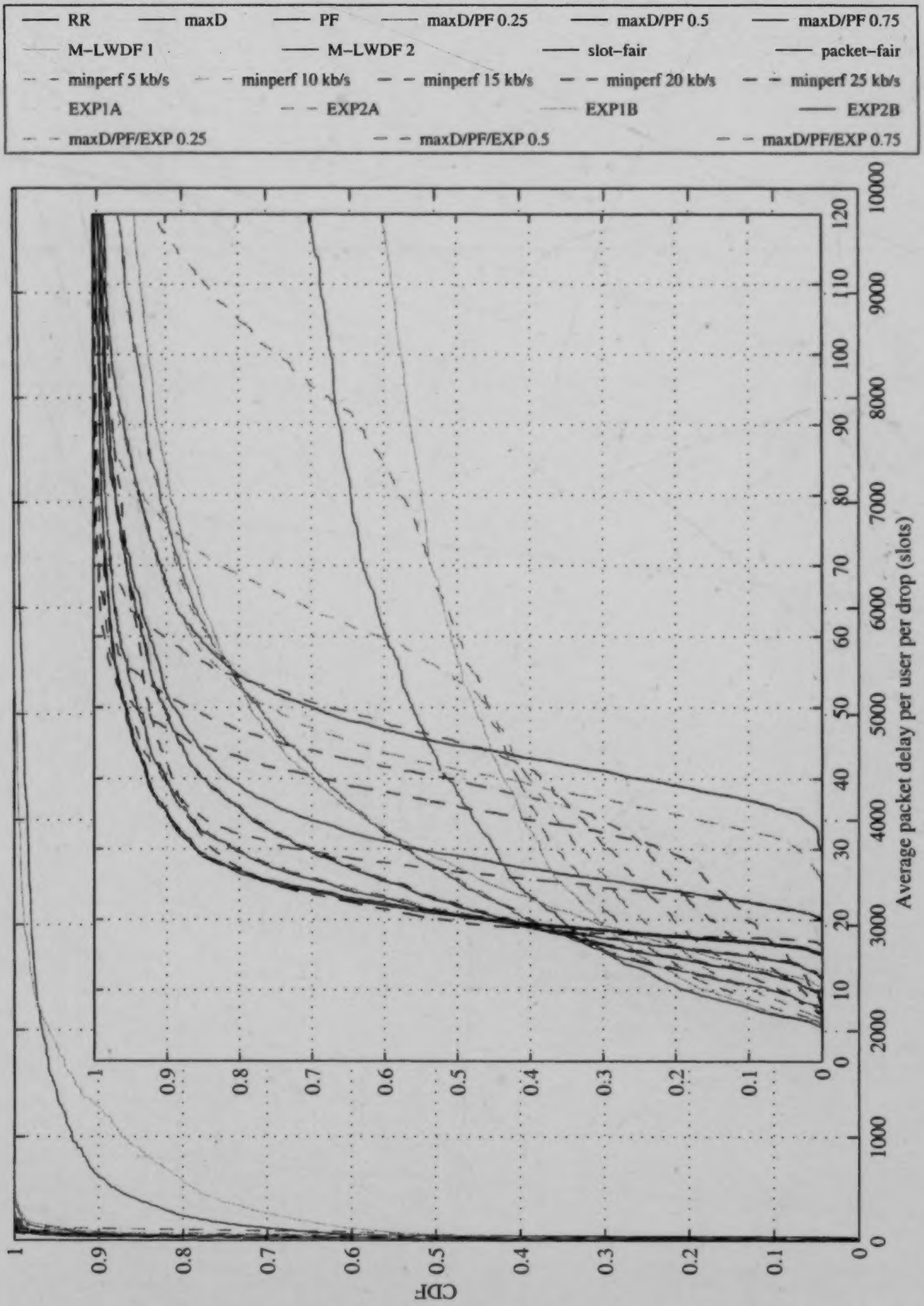


Figure C.17: Distributions of average packet delay per user for users who receive at least one packet in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction

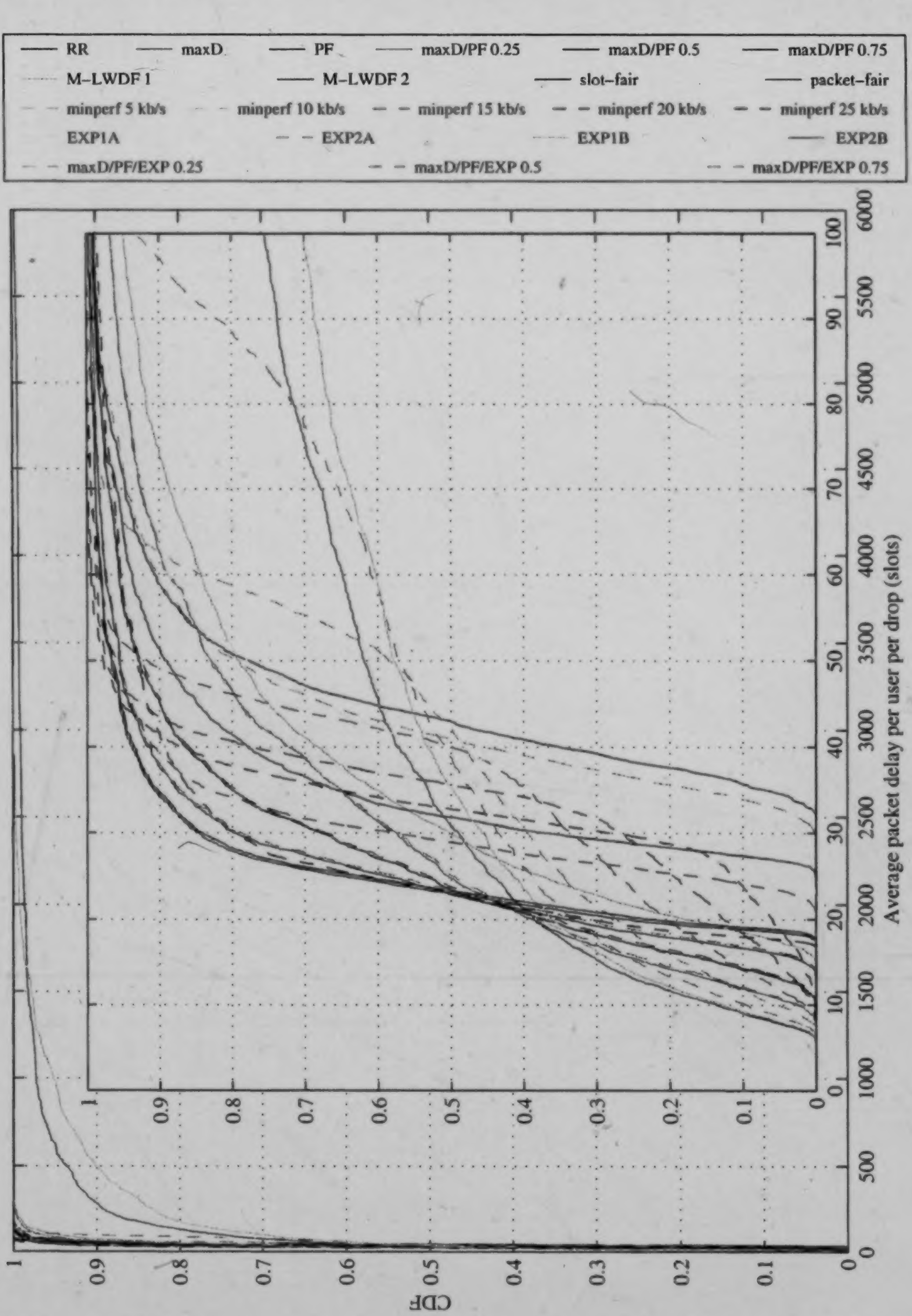


Figure C.18: Distributions of average packet delay per user for users who receive at least one packet in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction

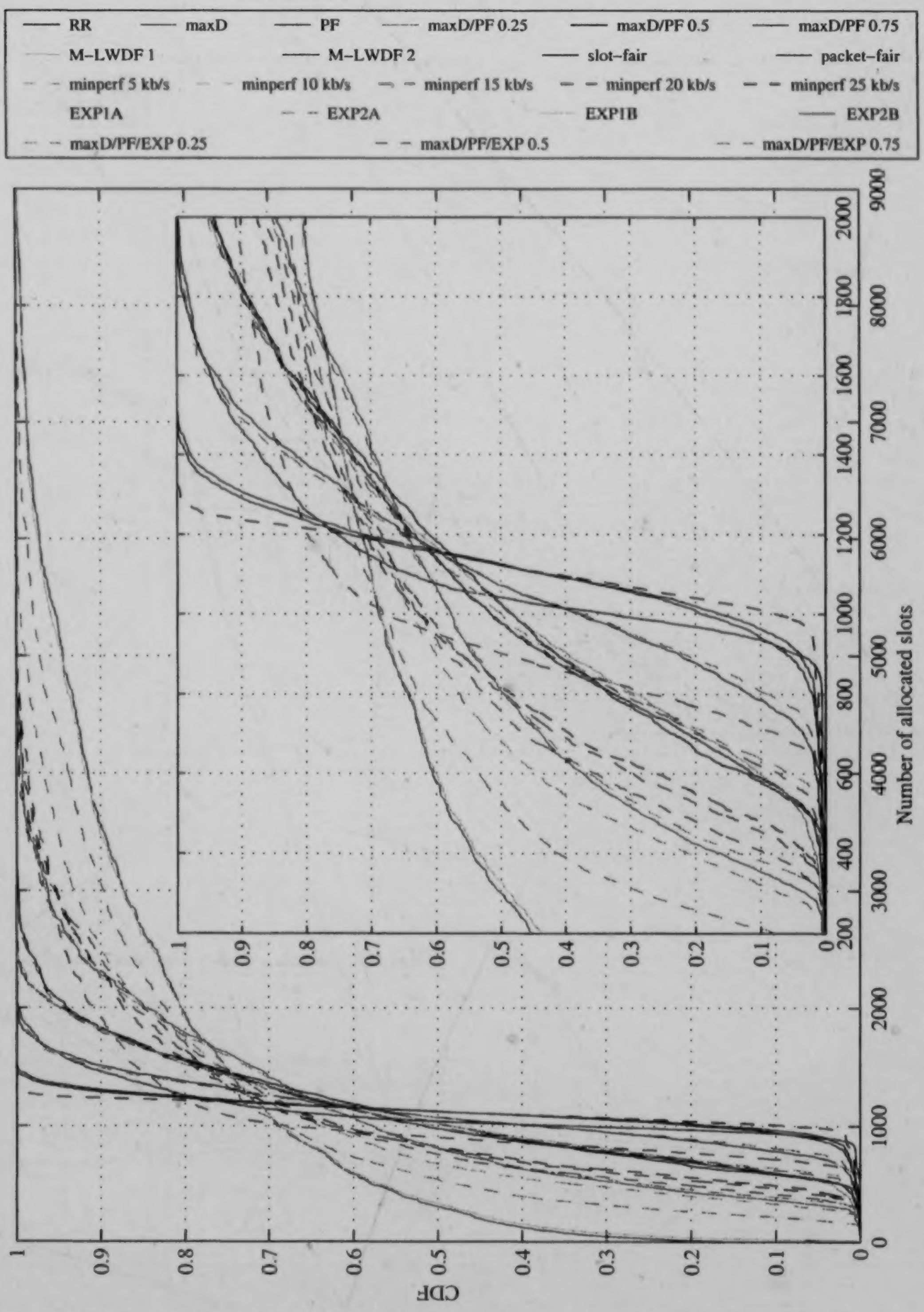


Figure C.19: Distributions of allocated slots per user per drop in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction

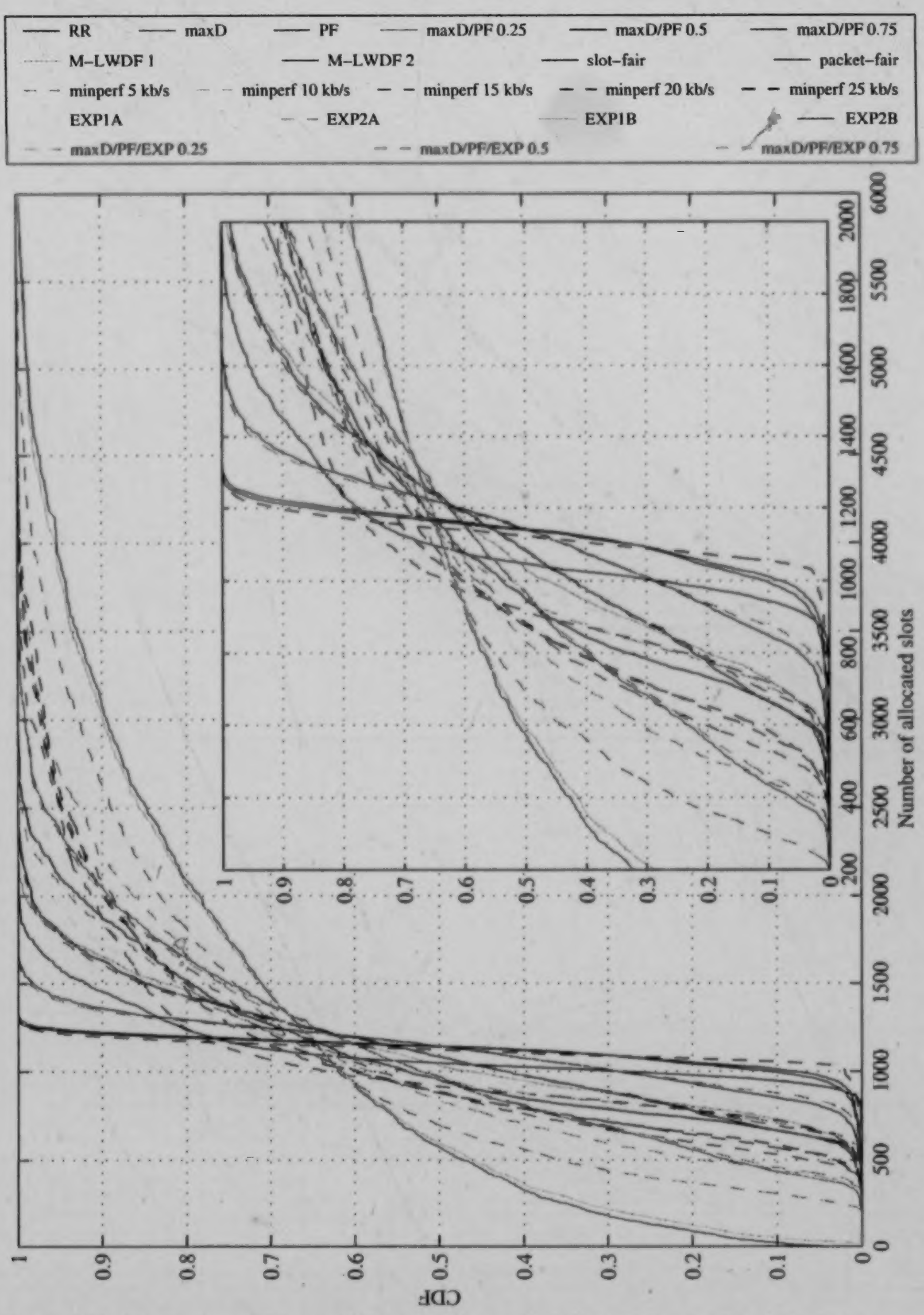


Figure C.20: Distributions of allocated slots per user per drop in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction

Appendix D: Pedestrian Channel Results Comparing the "PF A" and "PF B" Average Rate Calculation Methods using Perfect Multislot Prediction

This appendix contains graphs comparing the simulation results using the "PF A" and "PF B" methods for calculating the average rate in the metrics of scheduling algorithms for the ITU pedestrian A and B channels when perfect multislot prediction was used to determine the transmission format each user could support.

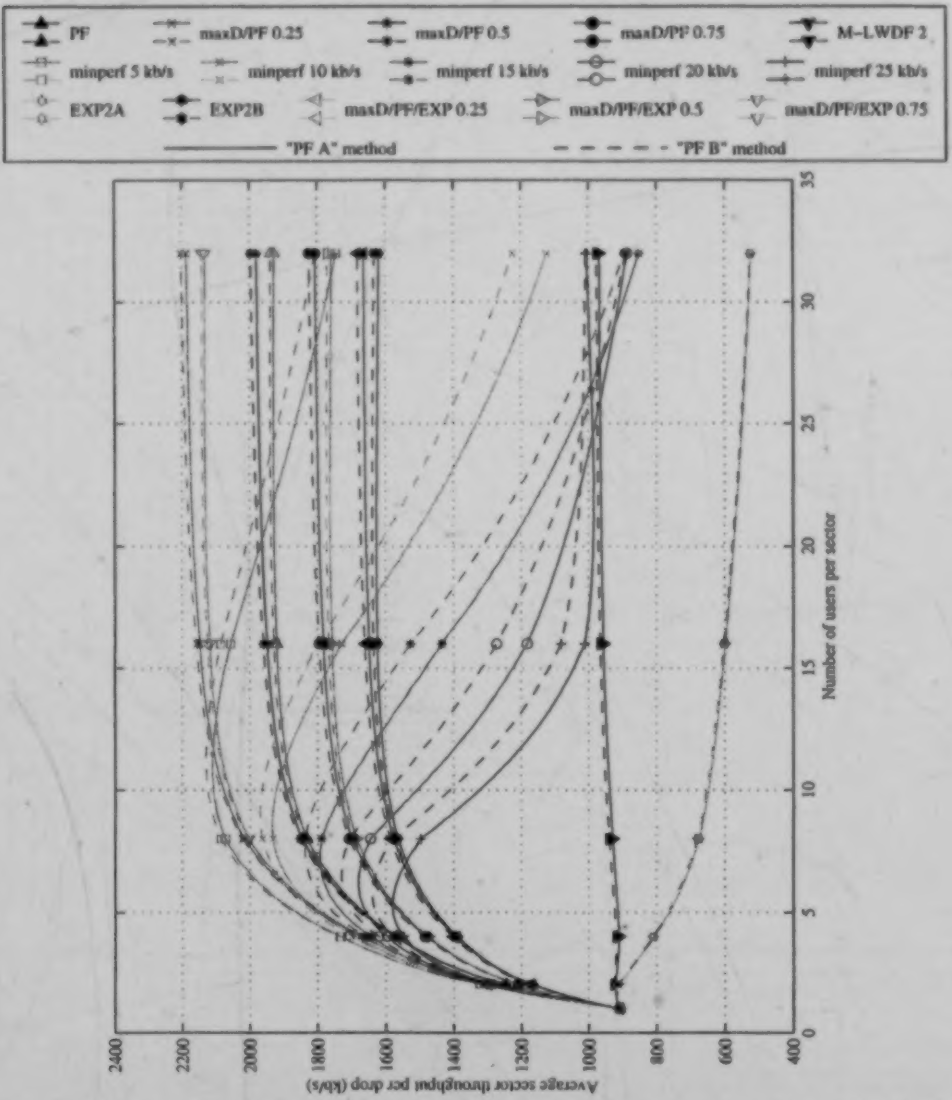


Figure D.1: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian A channel using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

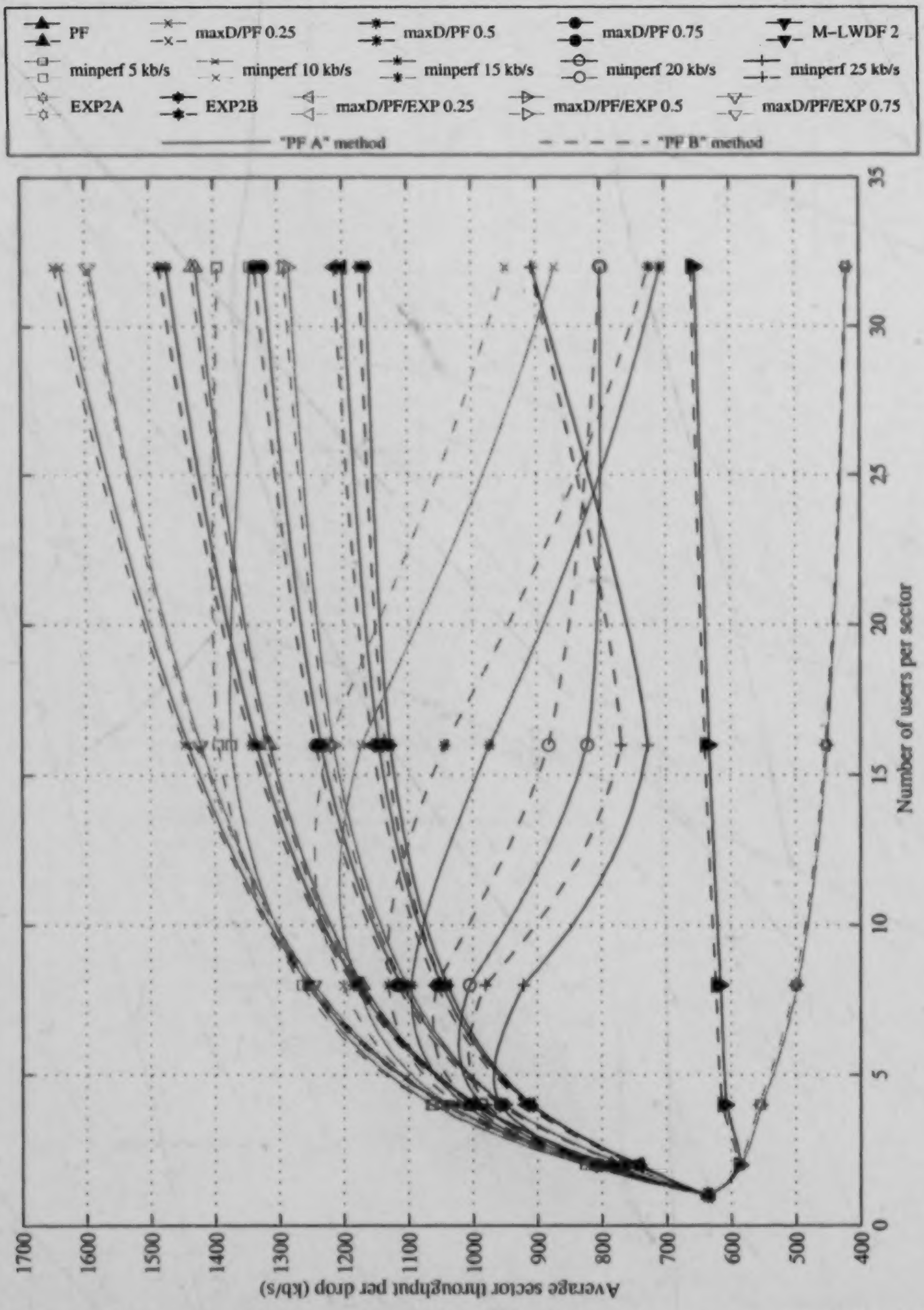


Figure D.2: Average sector throughput per drop vs. number of users per sector in the ITU pedestrian B channel using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

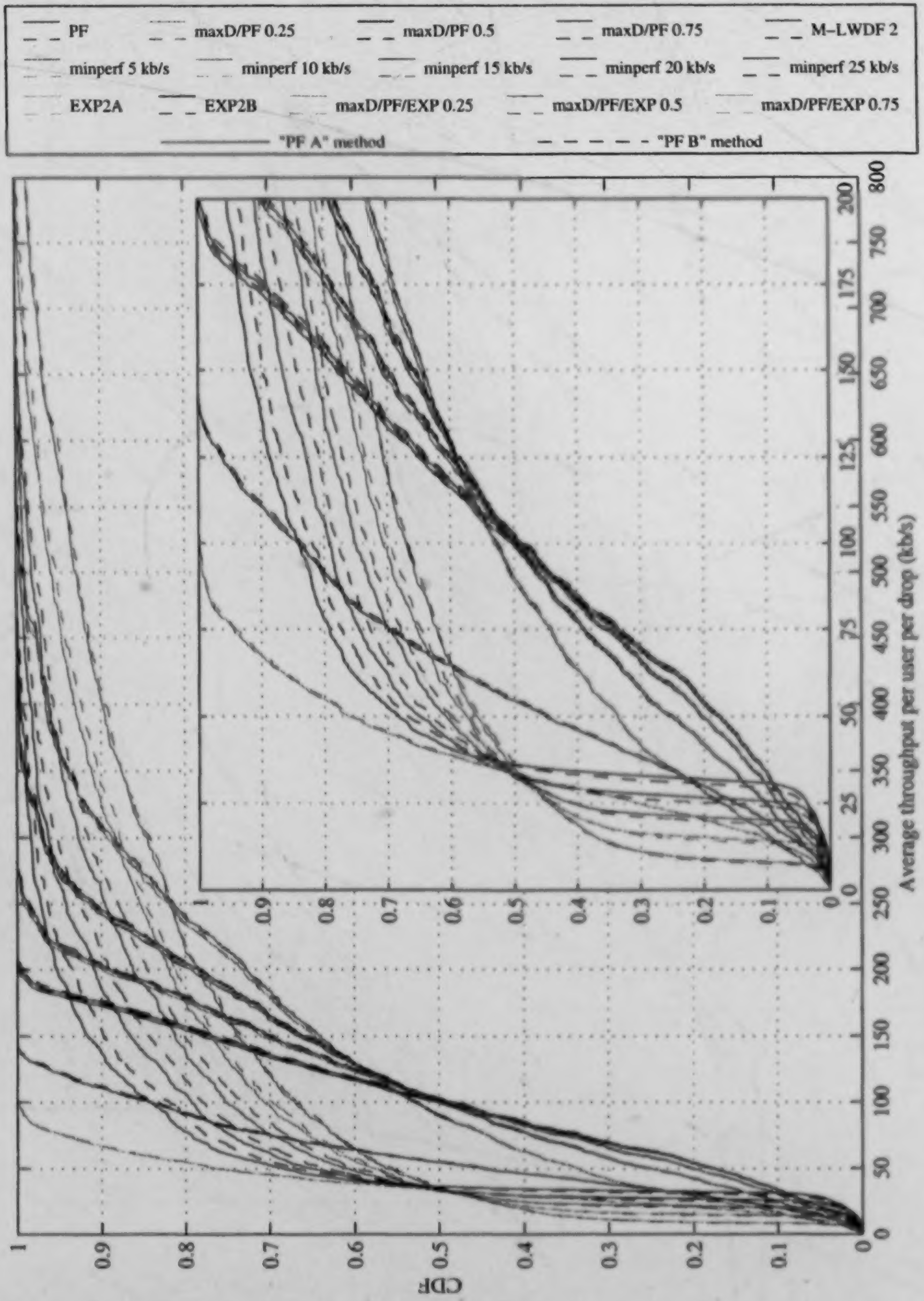


Figure D.3: Distributions of average throughput per user per drop in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

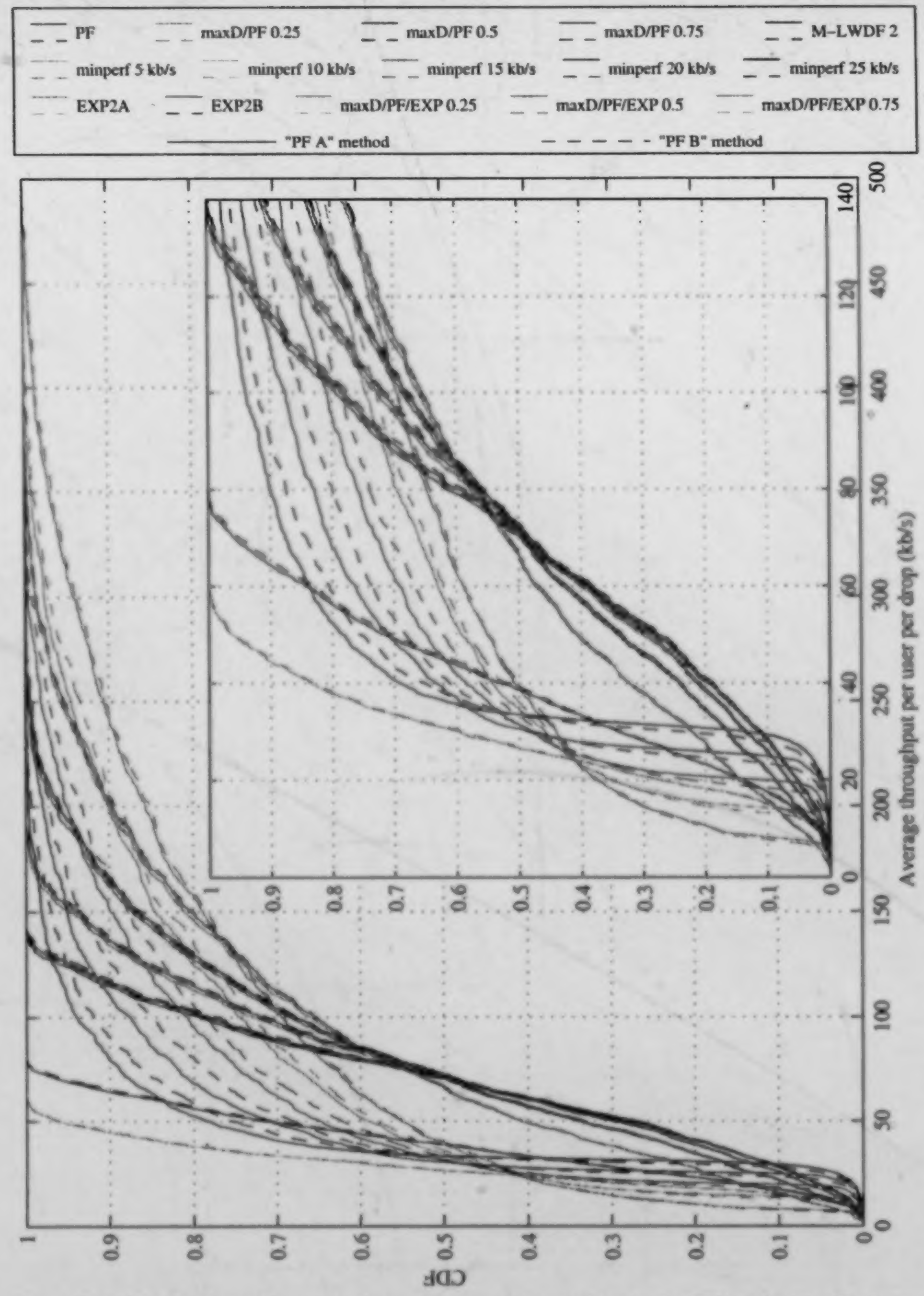


Figure D.4: Distributions of average throughput per user per drop in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

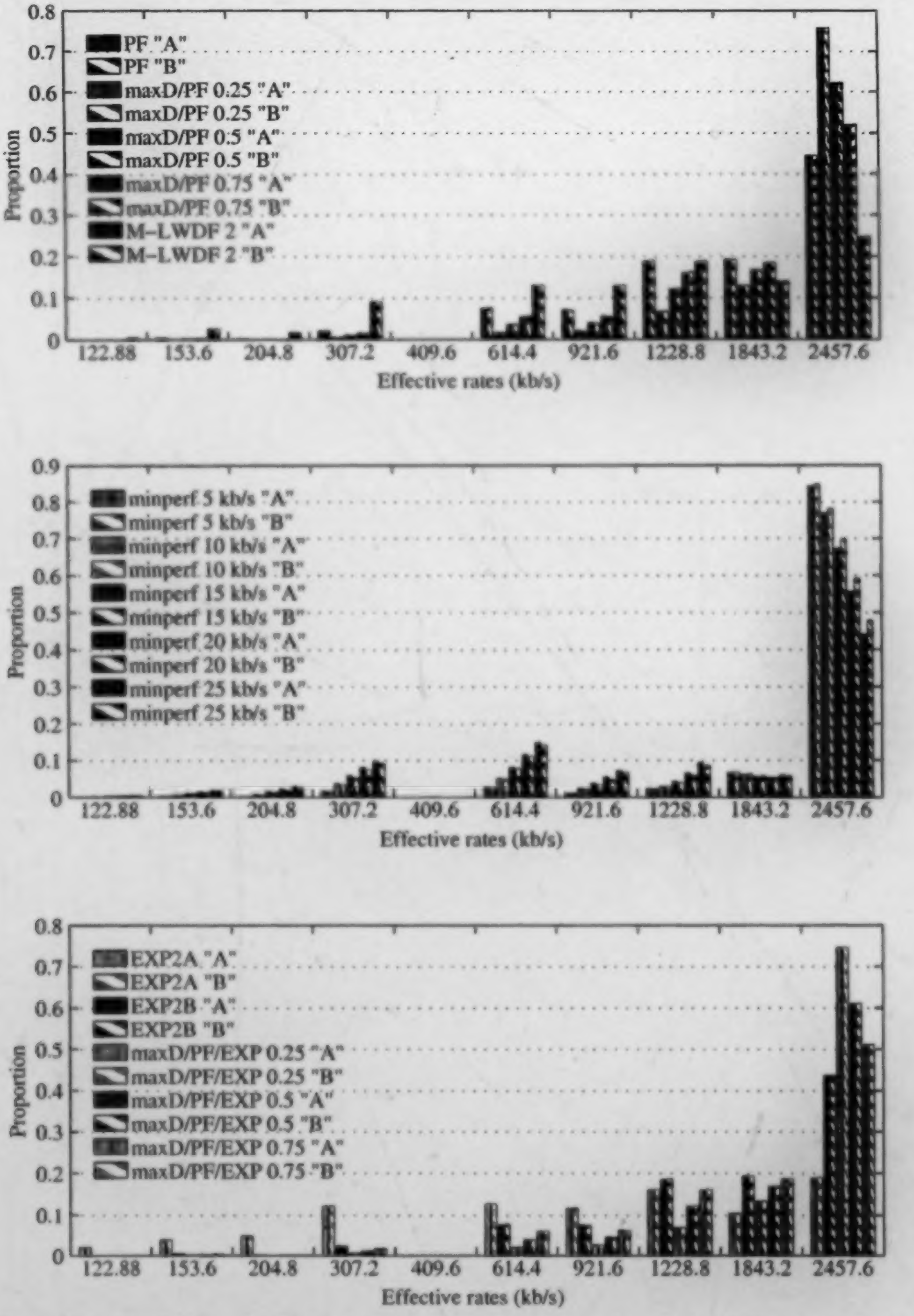


Figure D.5: Distribution of effective bit rates for transmitted packets in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

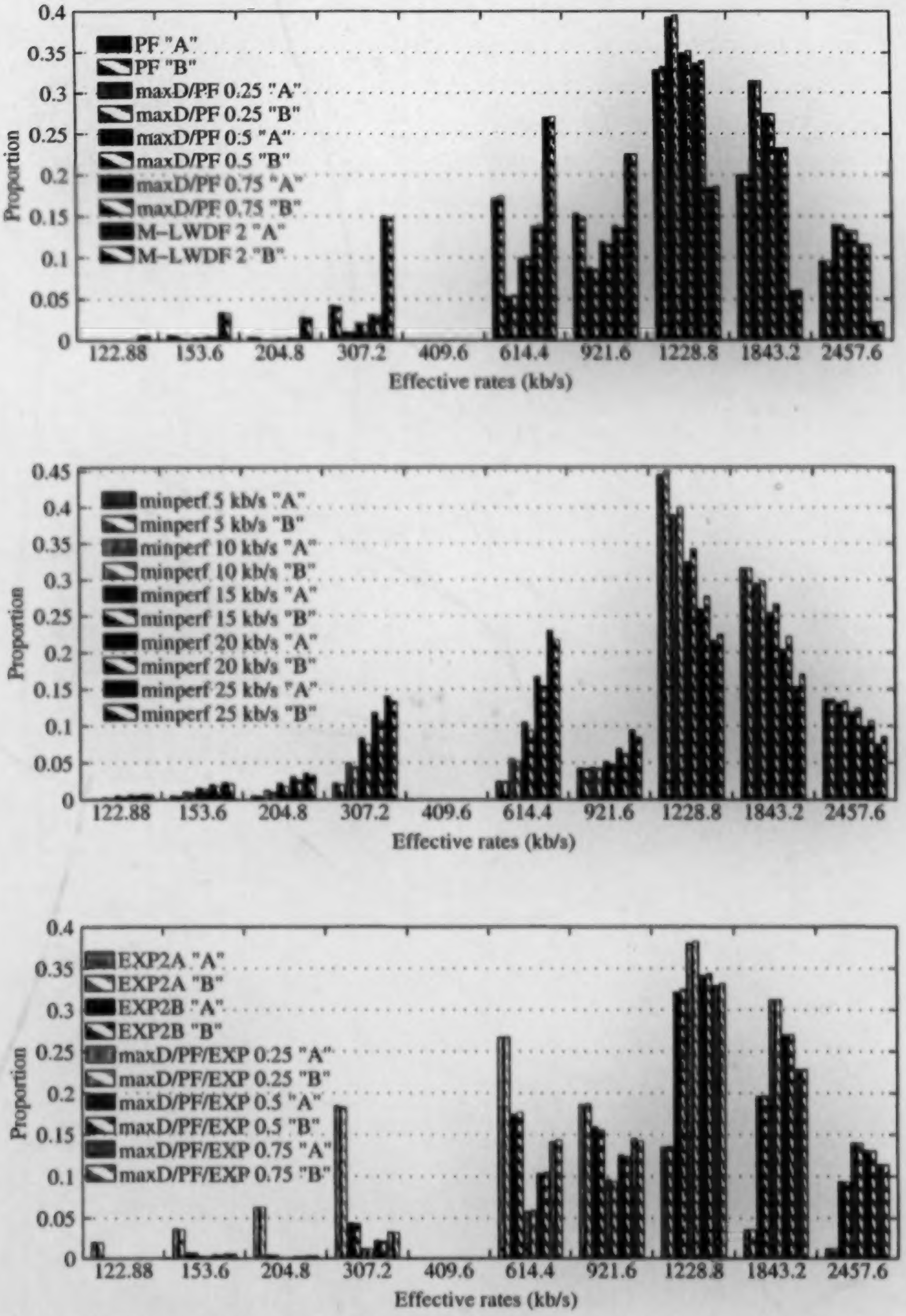


Figure D.6: Distribution of effective bit rates for transmitted packets in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

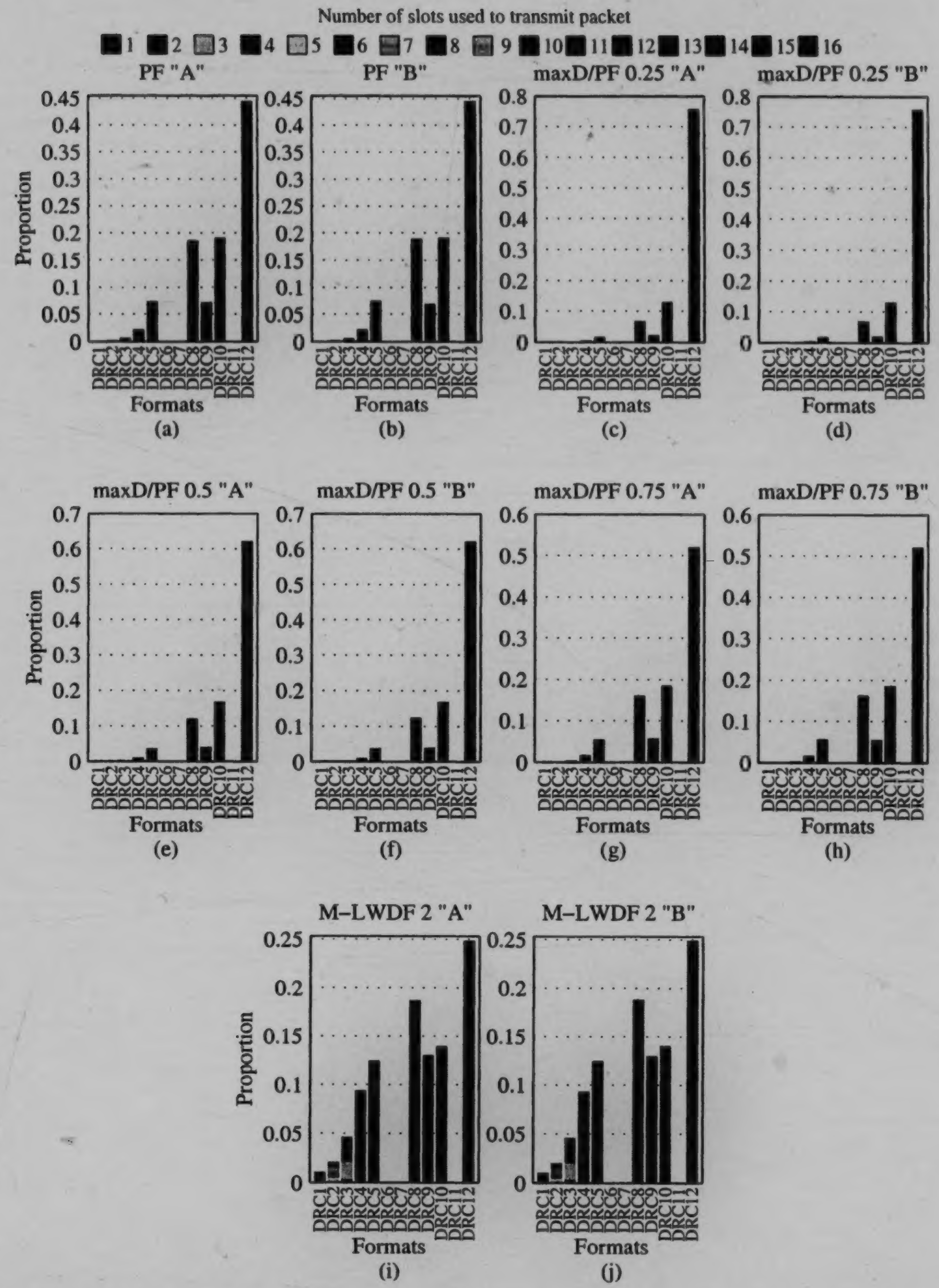


Figure D.7 (Part 1 of 3): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

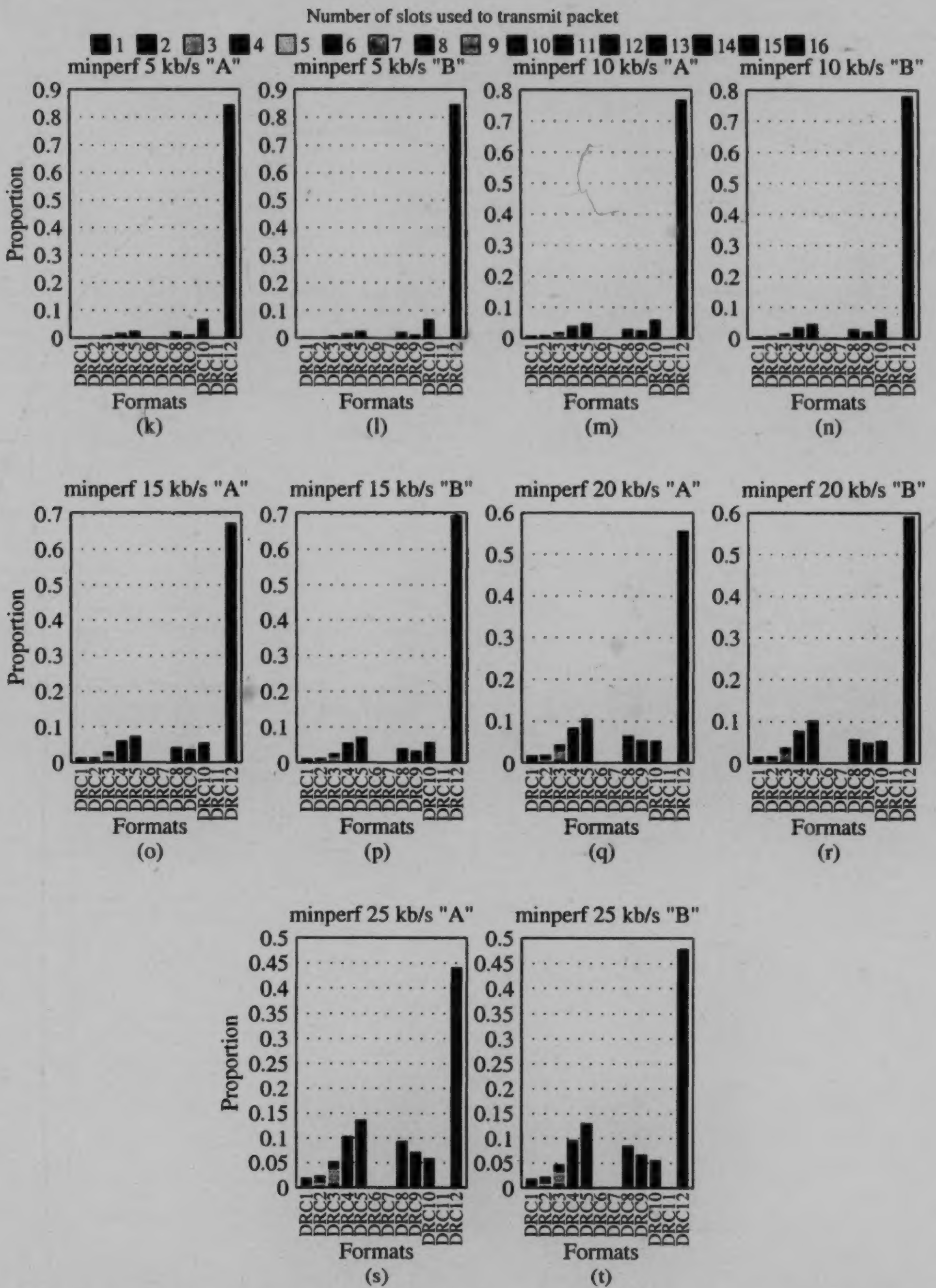


Figure D.7 (Part 2 of 3): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

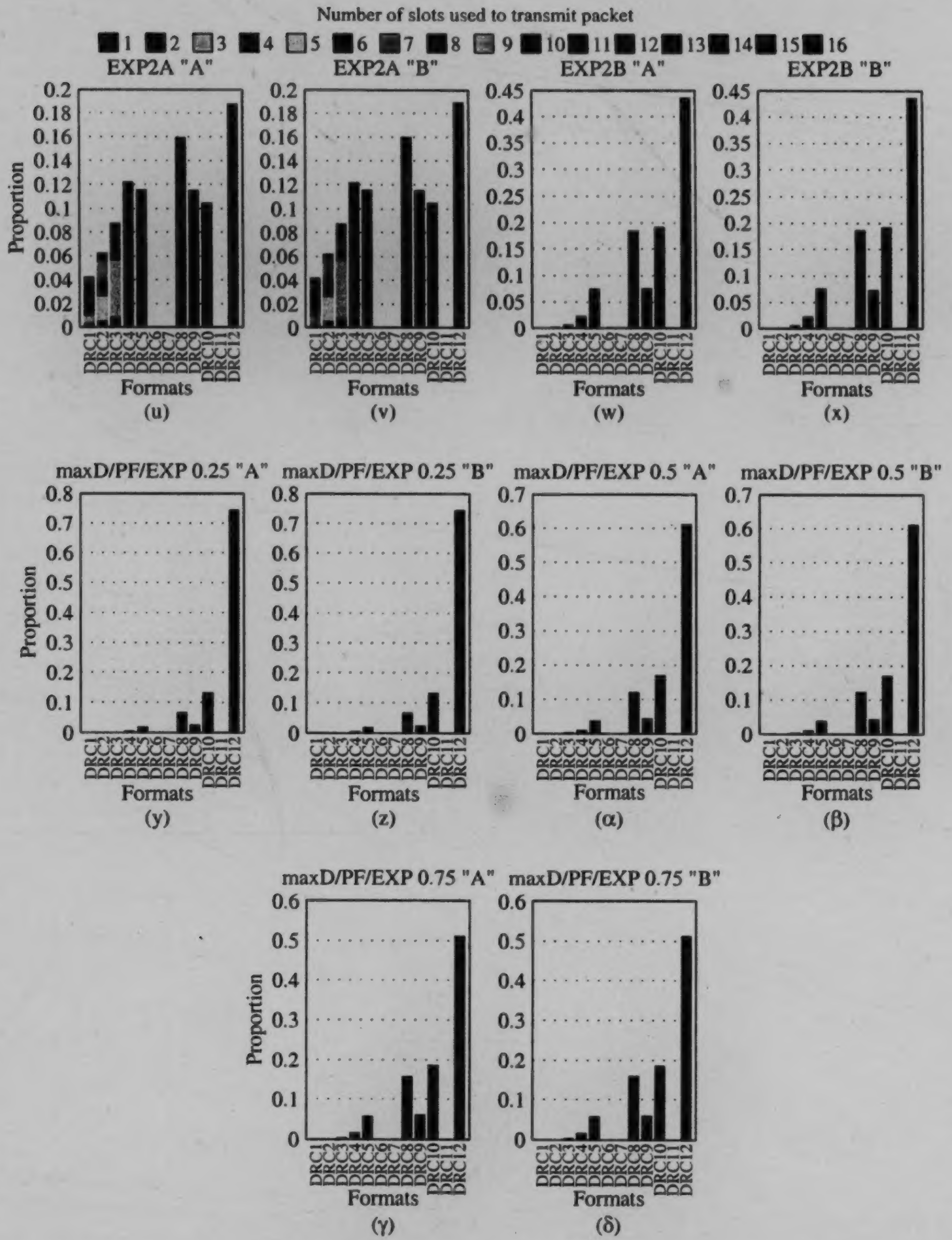


Figure D.7 (Part 3 of 3): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

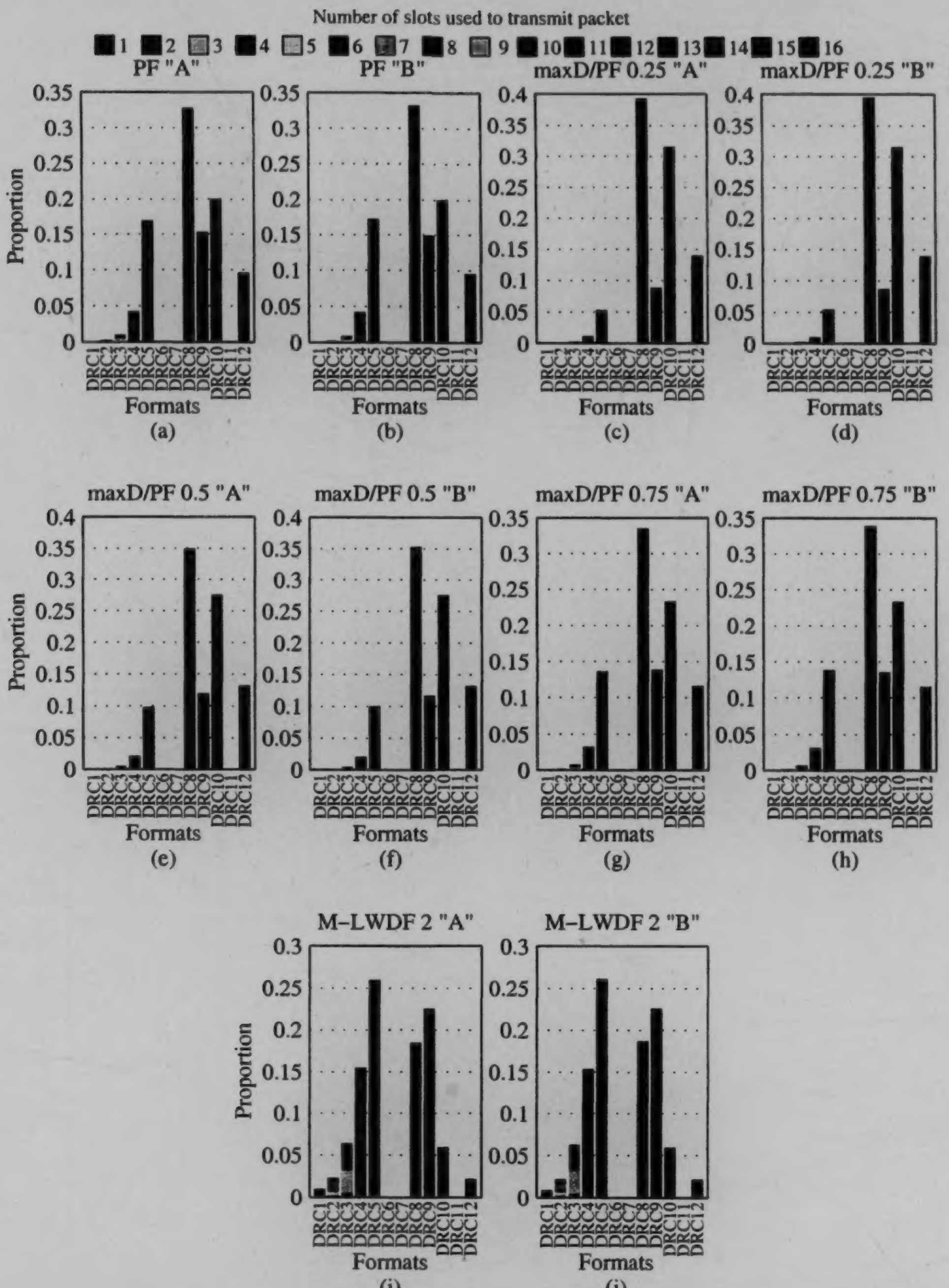
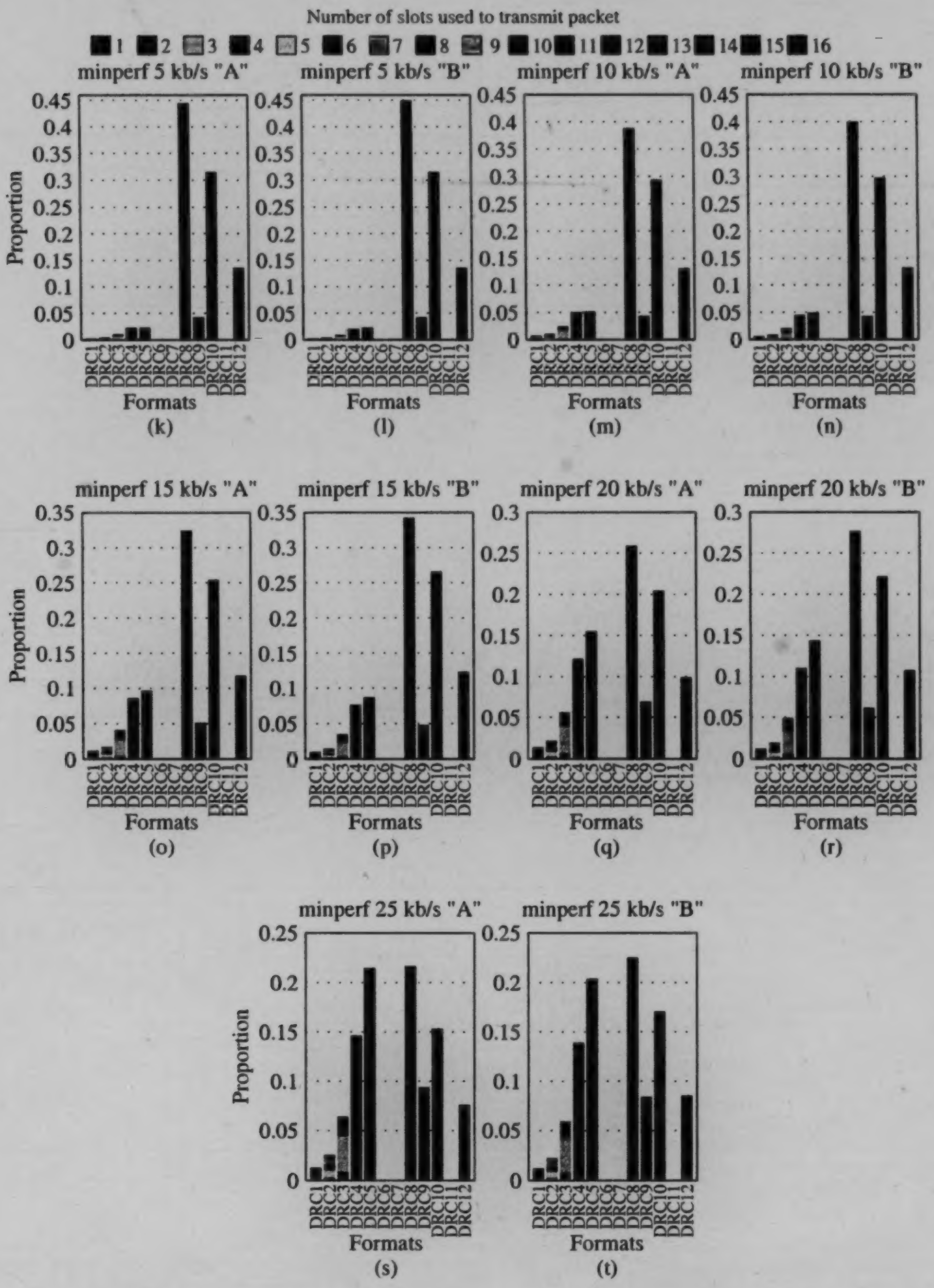


Figure D.8 (Part 1 of 3): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods



(k) (l) (m) (n)

(o) (p) (q) (r)

(s) (t)

Figure D.8 (Part 2 of 3): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction and comparing the “PF A” and “PF B” average rate calculation methods

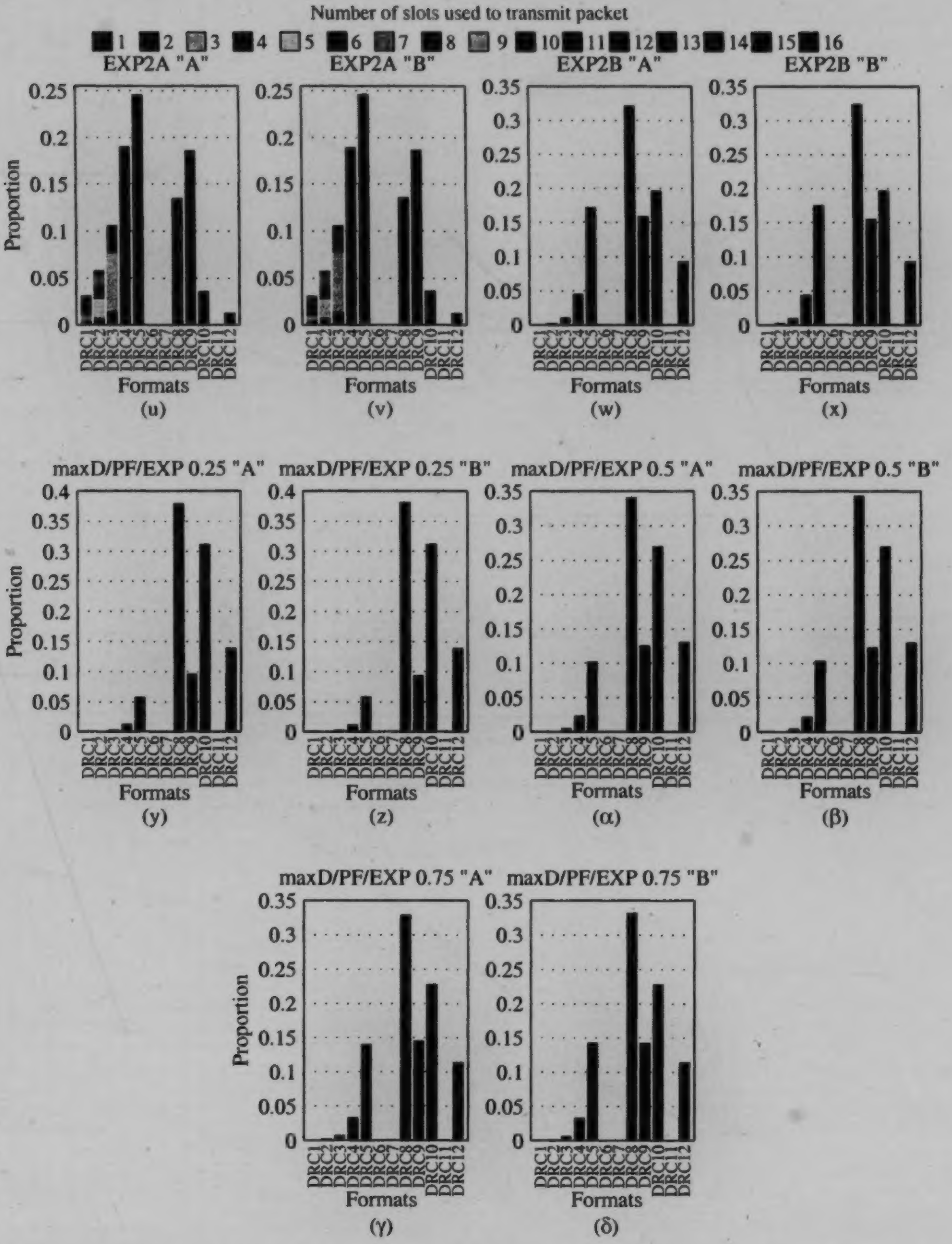


Figure D.8 (Part 3 of 3): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

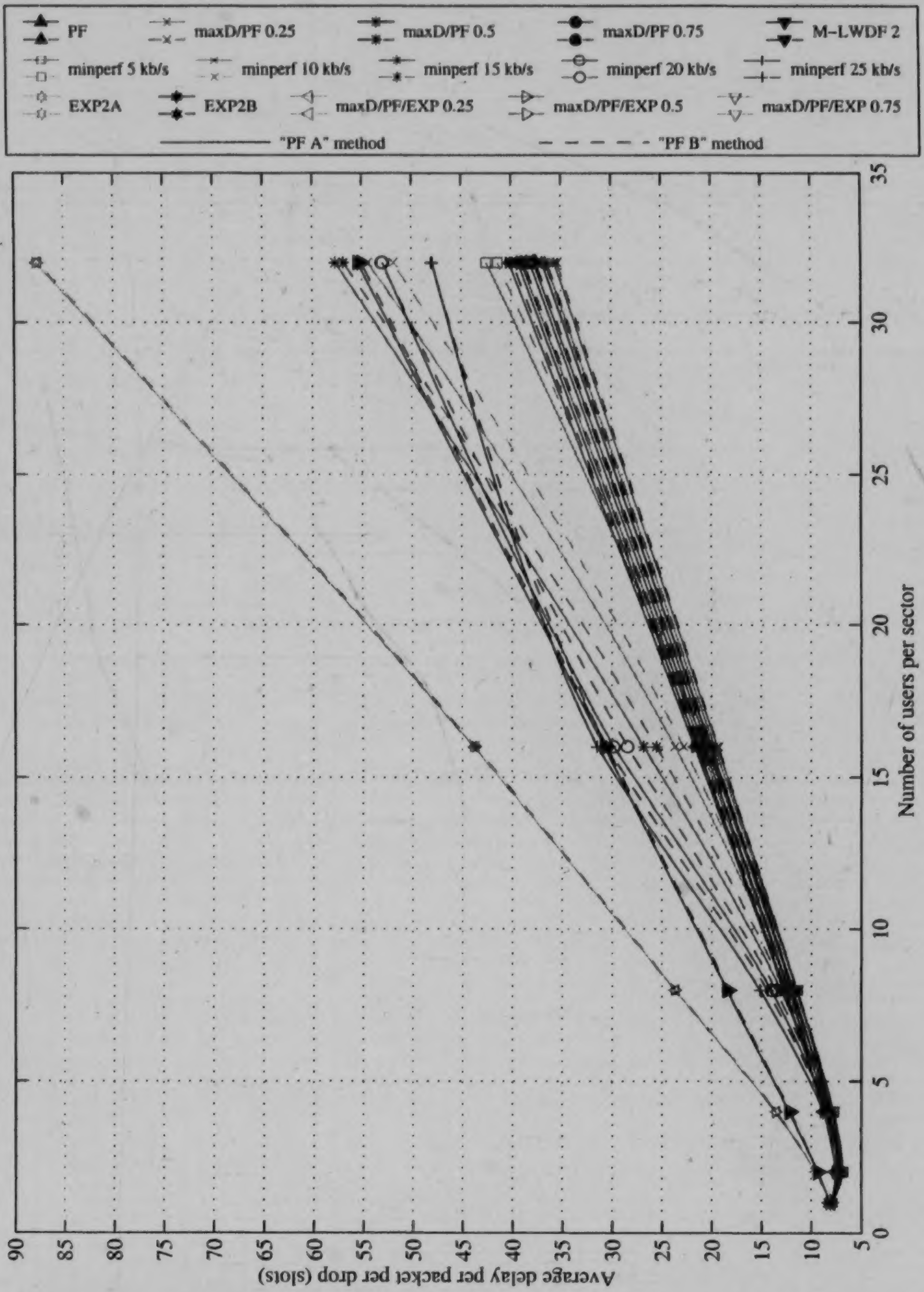


Figure D.9: Average delay per packet per drop versus number of users per sector in the ITU pedestrian A channel using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

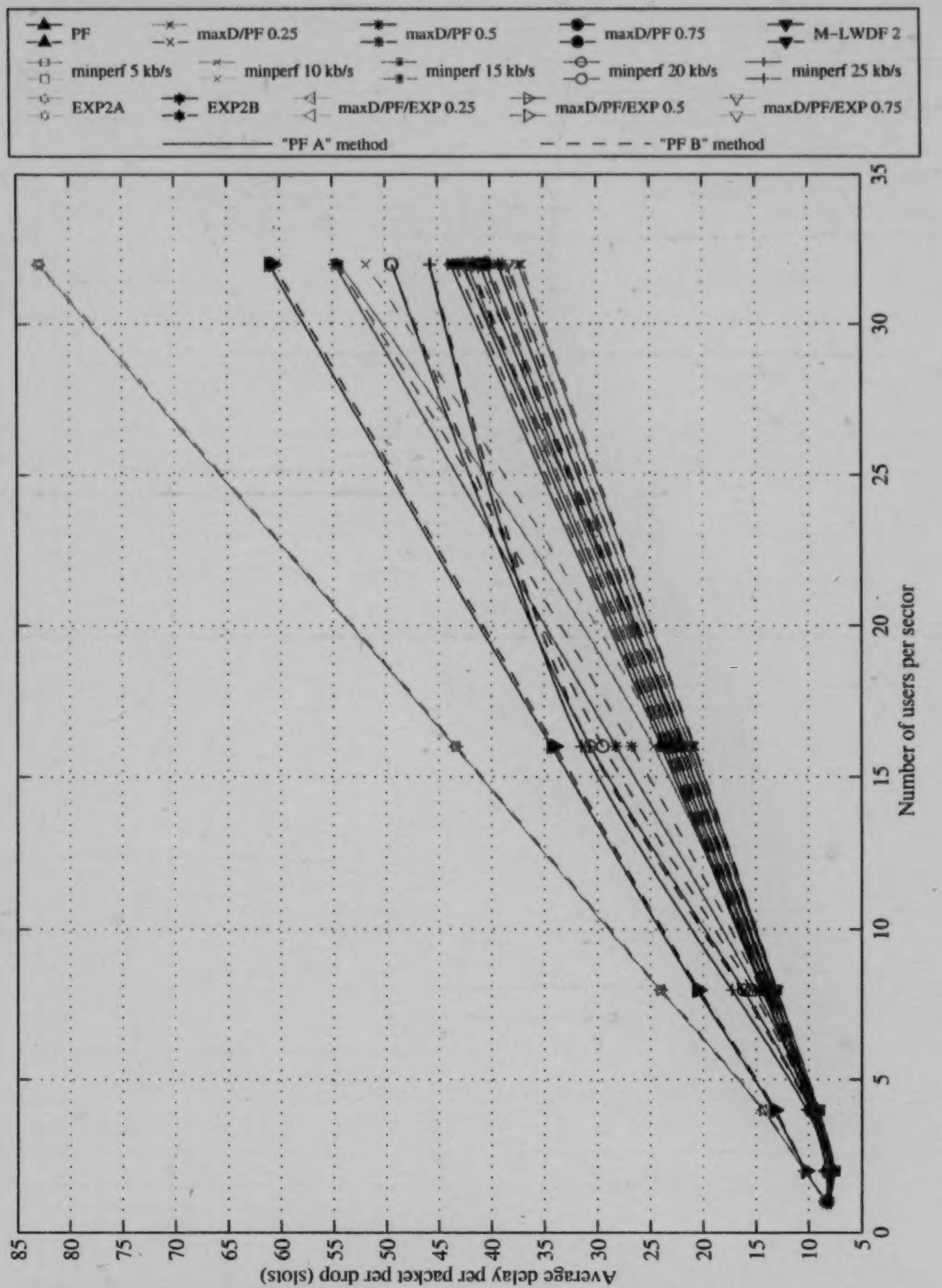


Figure D.10: Average delay per packet per drop versus number of users per sector in the ITU pedestrian B channel using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

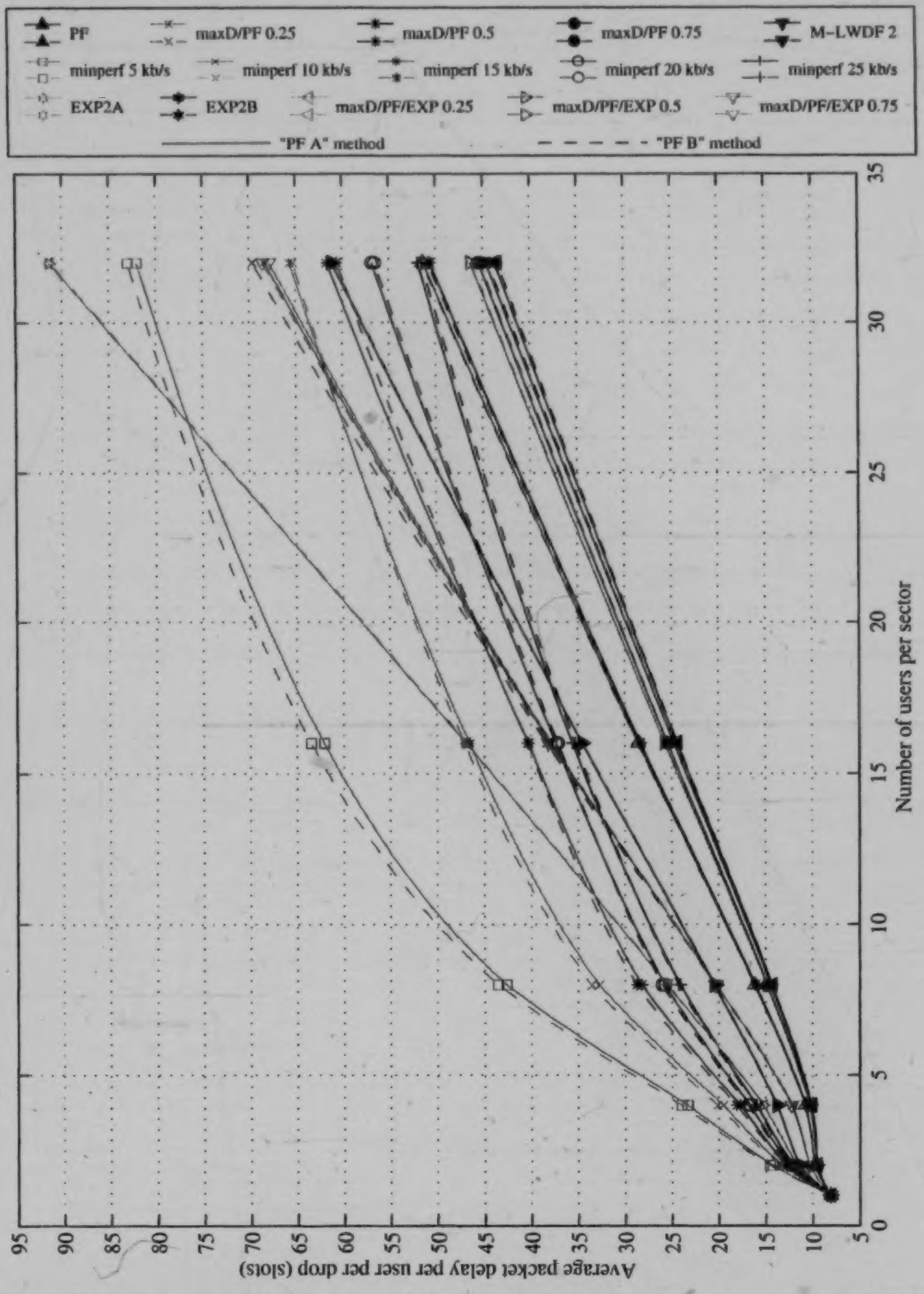


Figure D.11: Average packet delay per user per drop versus number of users per sector in the ITU pedestrian A channel using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

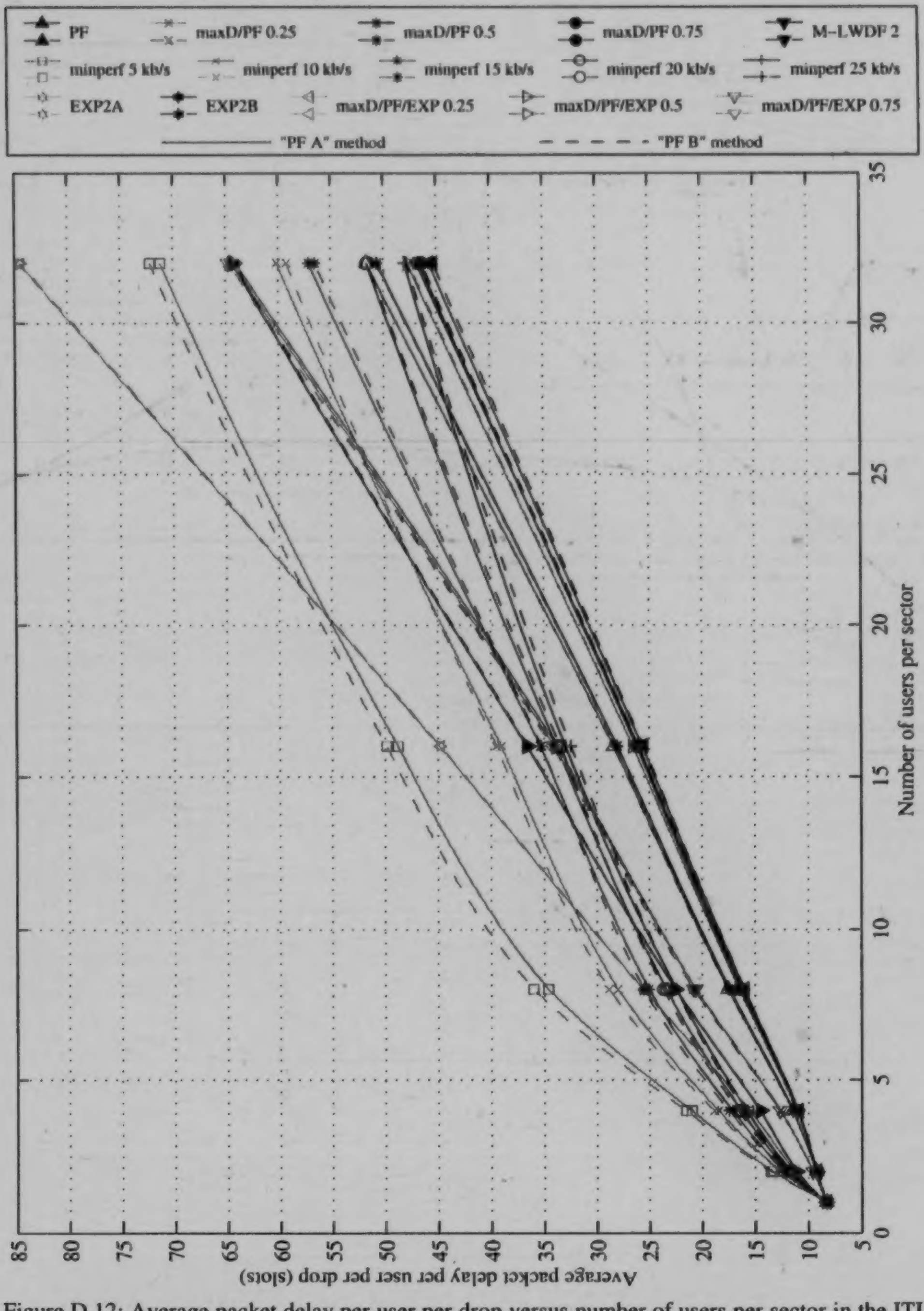


Figure D.12: Average packet delay per user per drop versus number of users per sector in the ITU pedestrian B channel using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

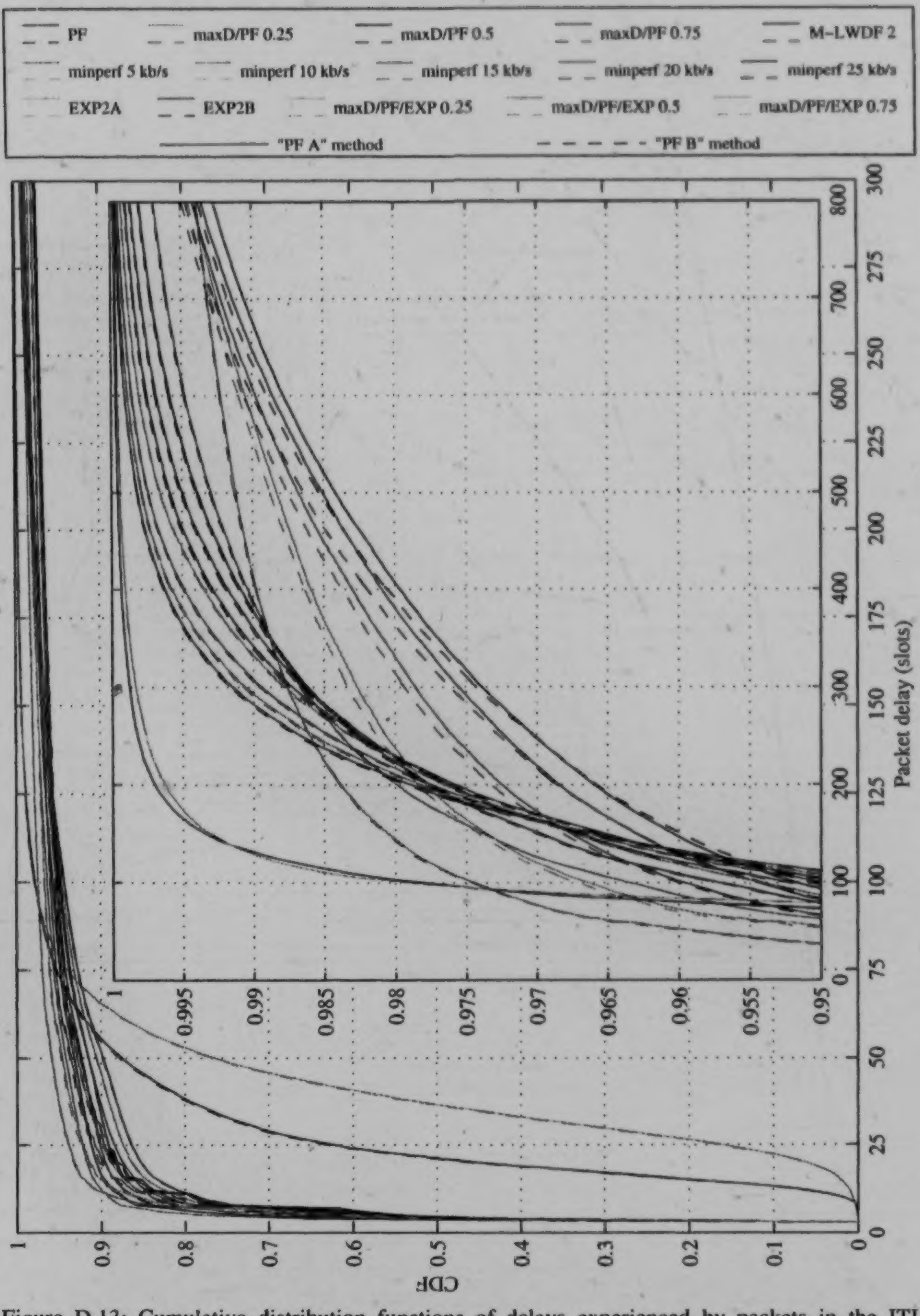


Figure D.13: Cumulative distribution functions of delays experienced by packets in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

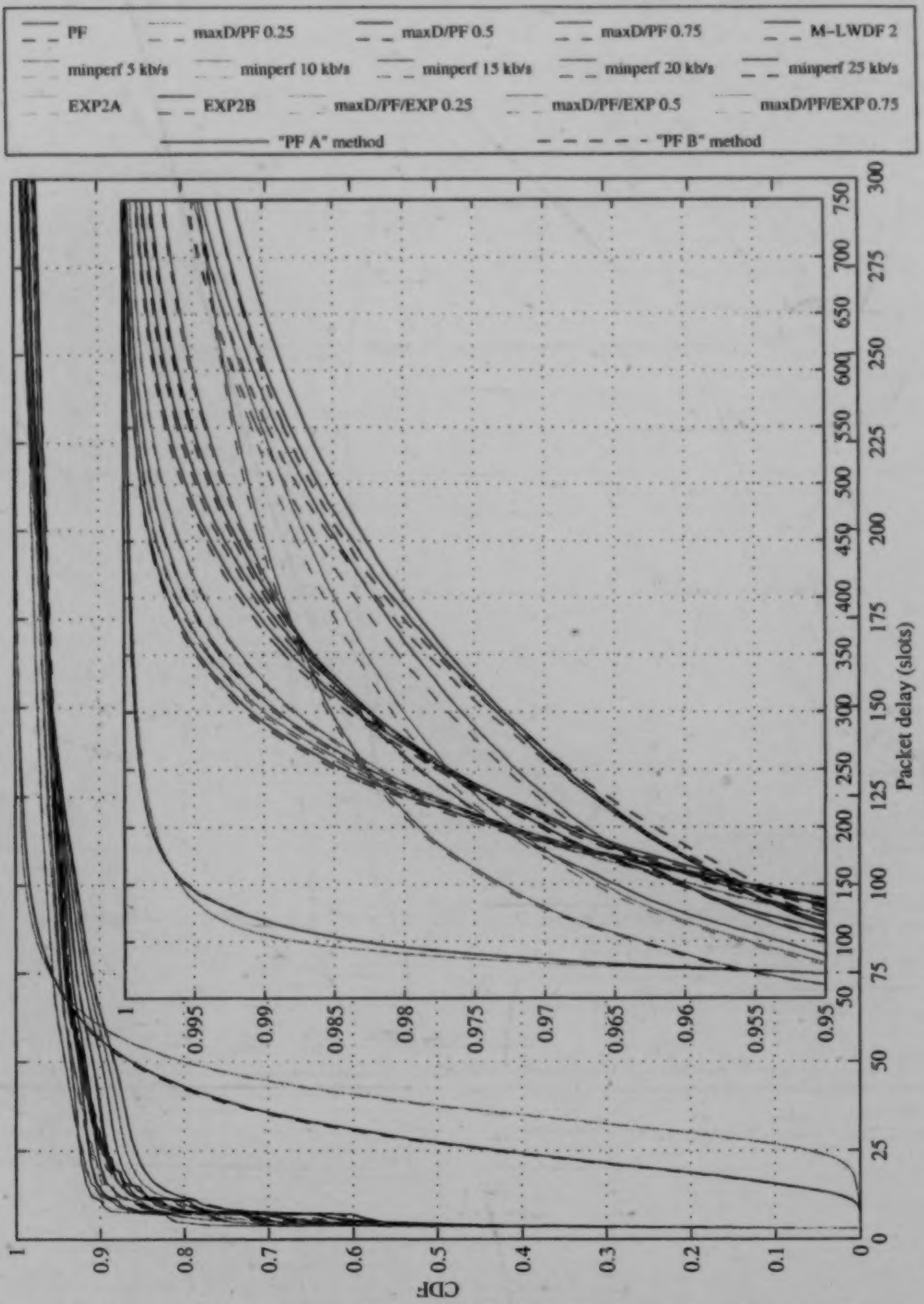


Figure D.14: Cumulative distribution functions of delays experienced by packets in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

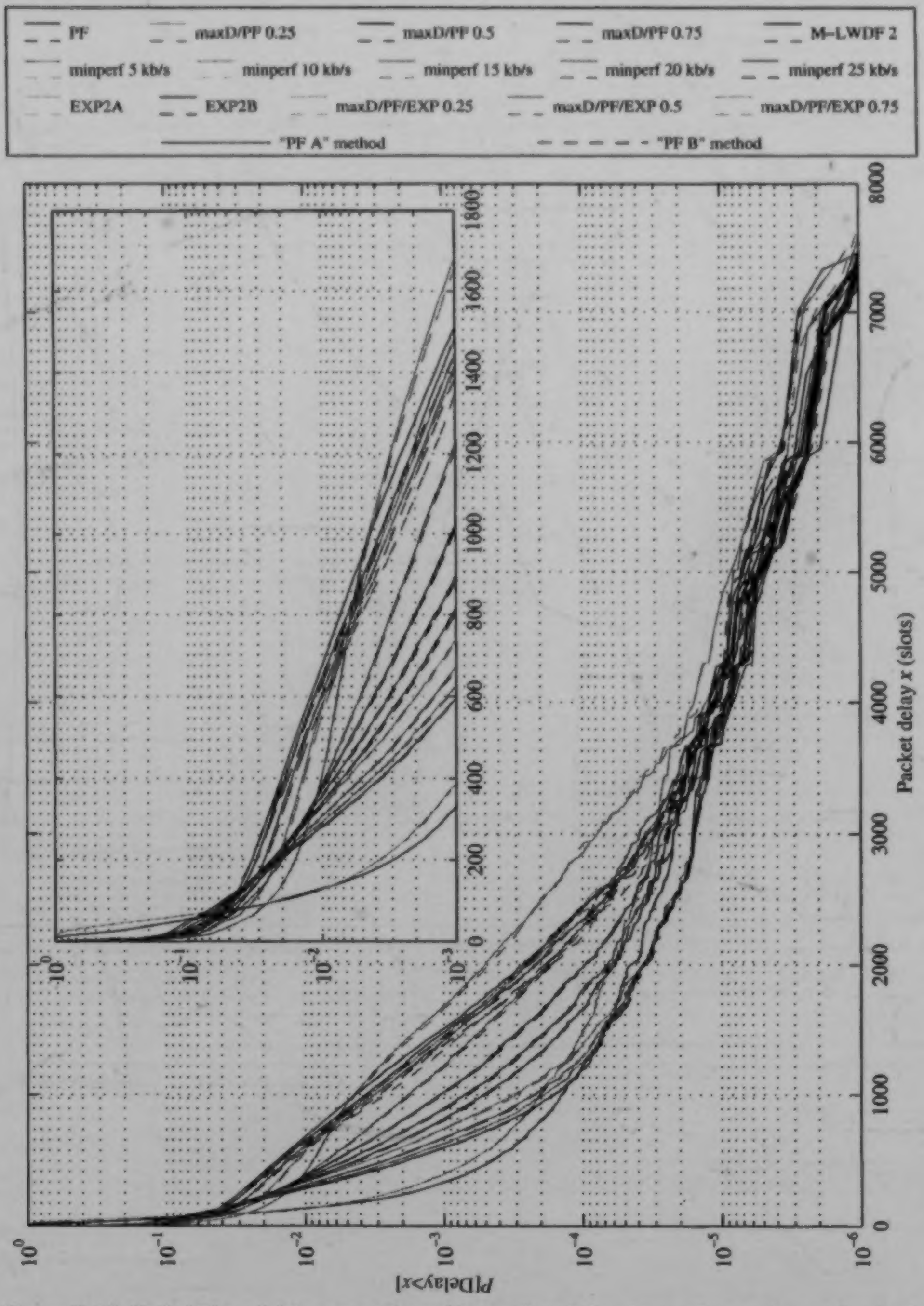


Figure D.15: Probability of delays experienced by packets being larger than a given value in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

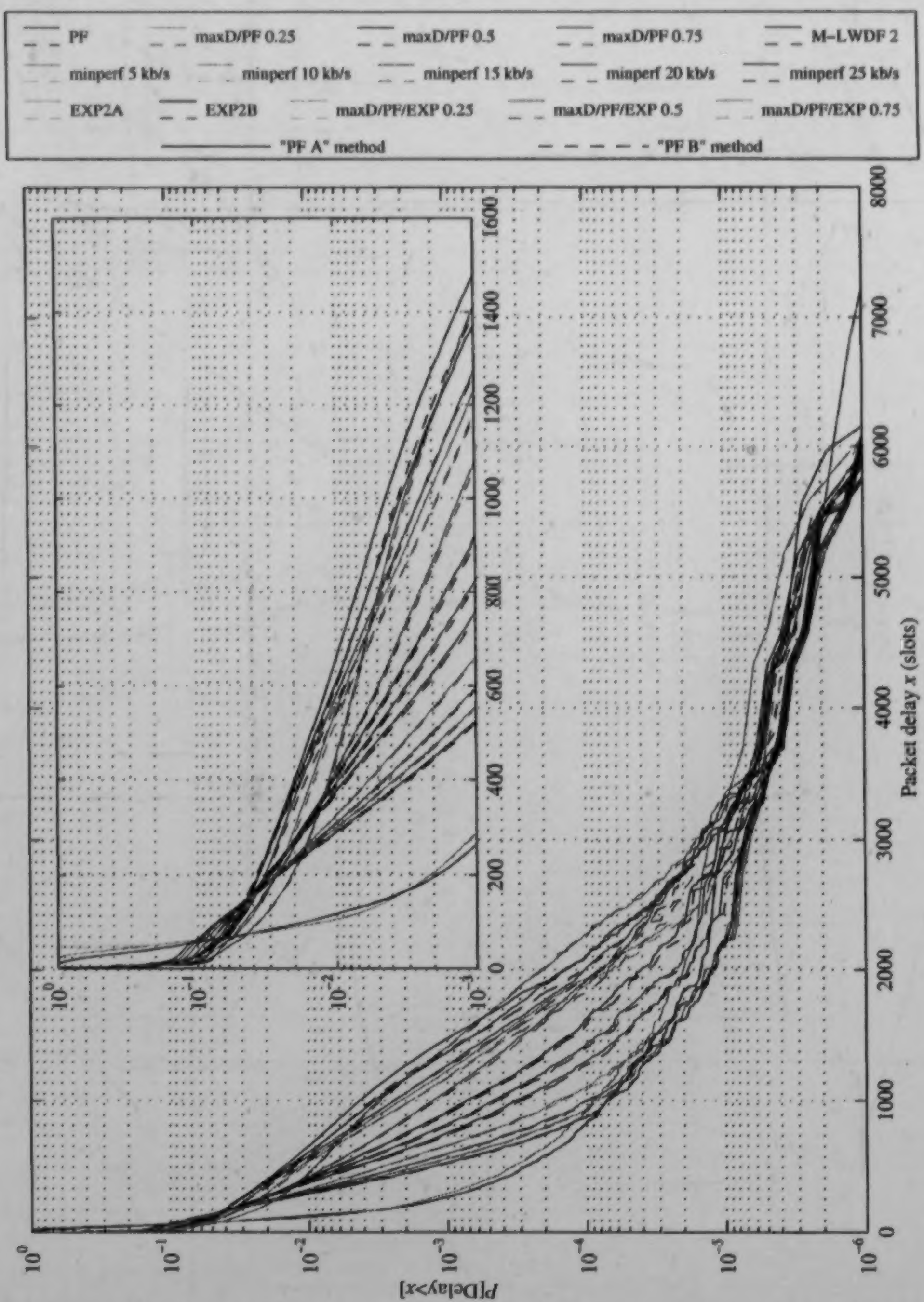


Figure D.16: Probability of delays experienced by packets being larger than a given value in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

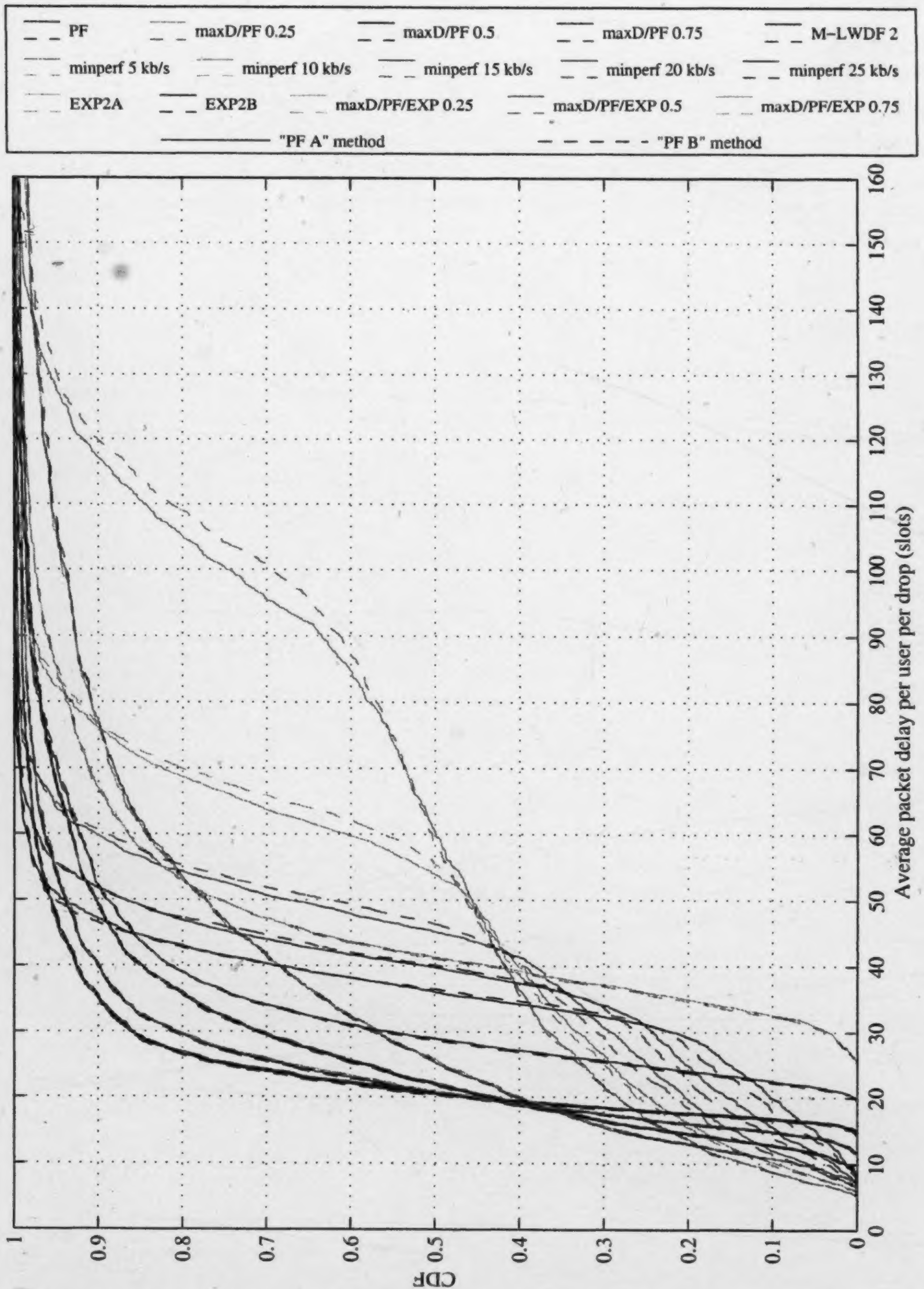


Figure D.17: Distributions of average packet delay per user for users who receive at least one packet in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

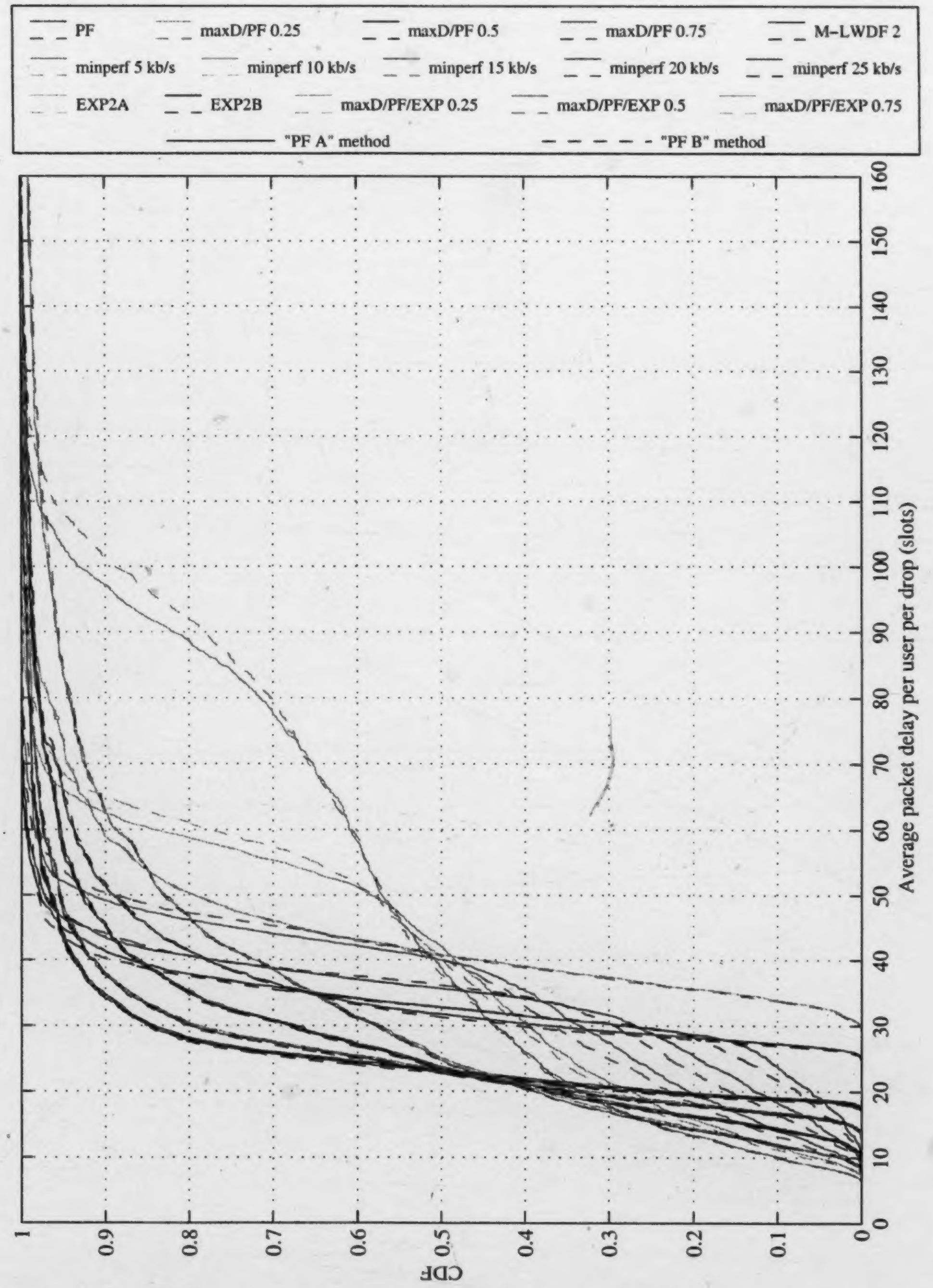


Figure D.18: Distributions of average packet delay per user for users who receive at least one packet in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

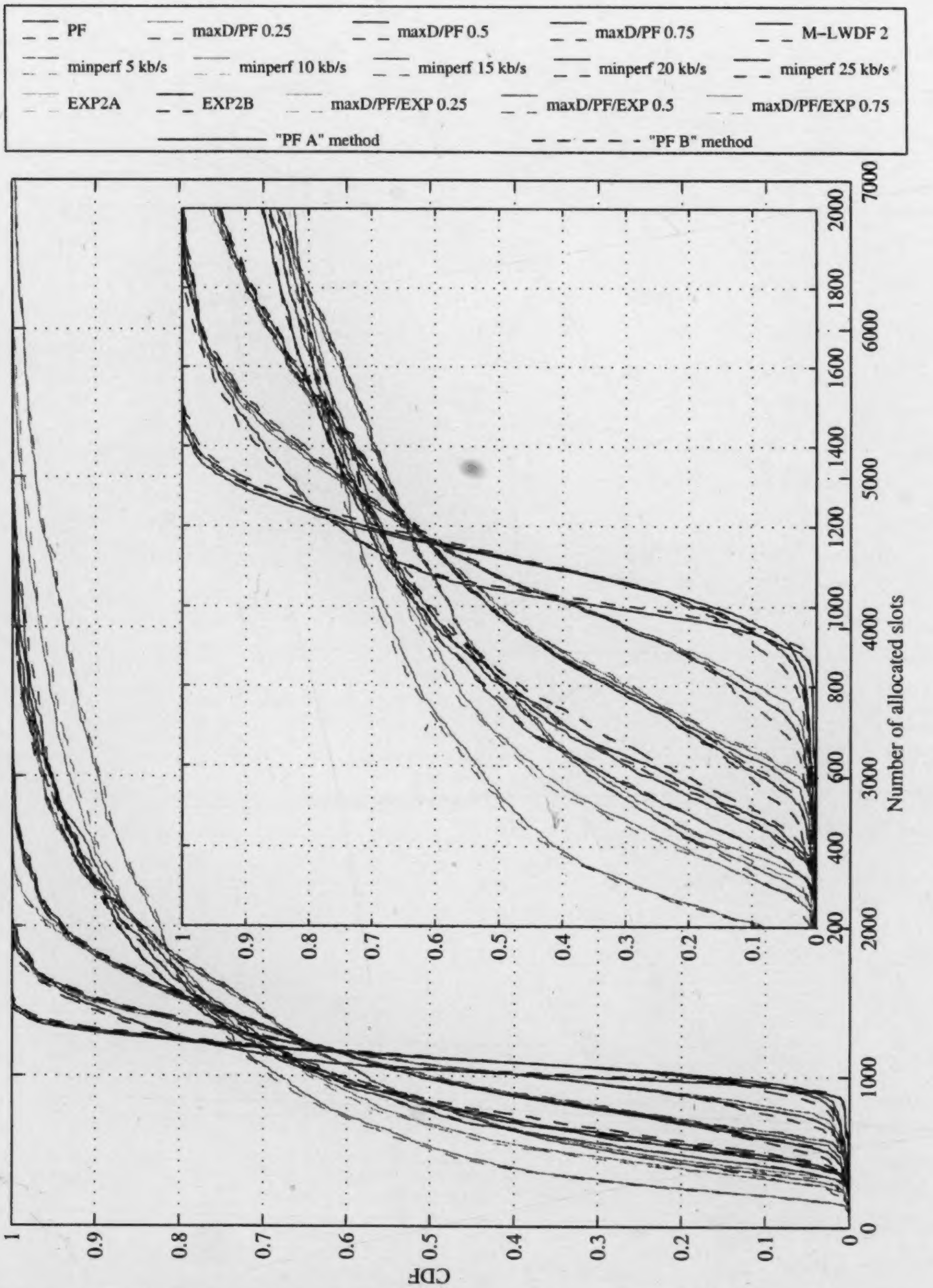


Figure D.19: Distributions of allocated slots per user per drop in the ITU pedestrian A channel with 16 users per sector using perfect multislot prediction and comparing the “PF A” and “PF B” average rate calculation methods

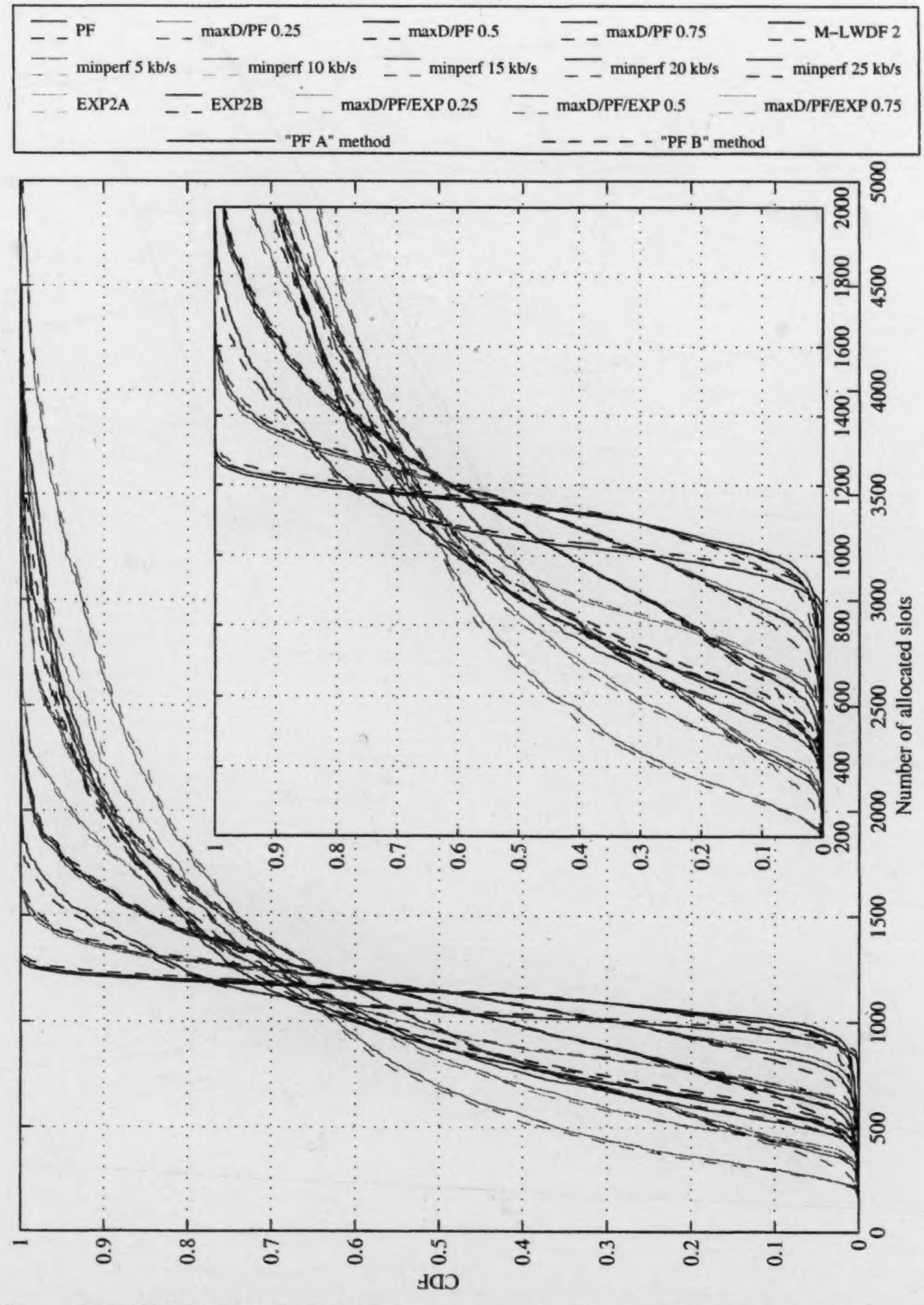


Figure D.20: Distributions of allocated slots per user per drop in the ITU pedestrian B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

Appendix E: ITU Vehicular B Channel Results using Outdated SIR$_p$ Values and Margins

This appendix contains graphs showing the simulation results for the ITU vehicular B channel when outdated SIR$_p$ values (i.e. the signal-to-interference ratio minus the Rayleigh component) plus margins were used to determine the transmission format each user could support.

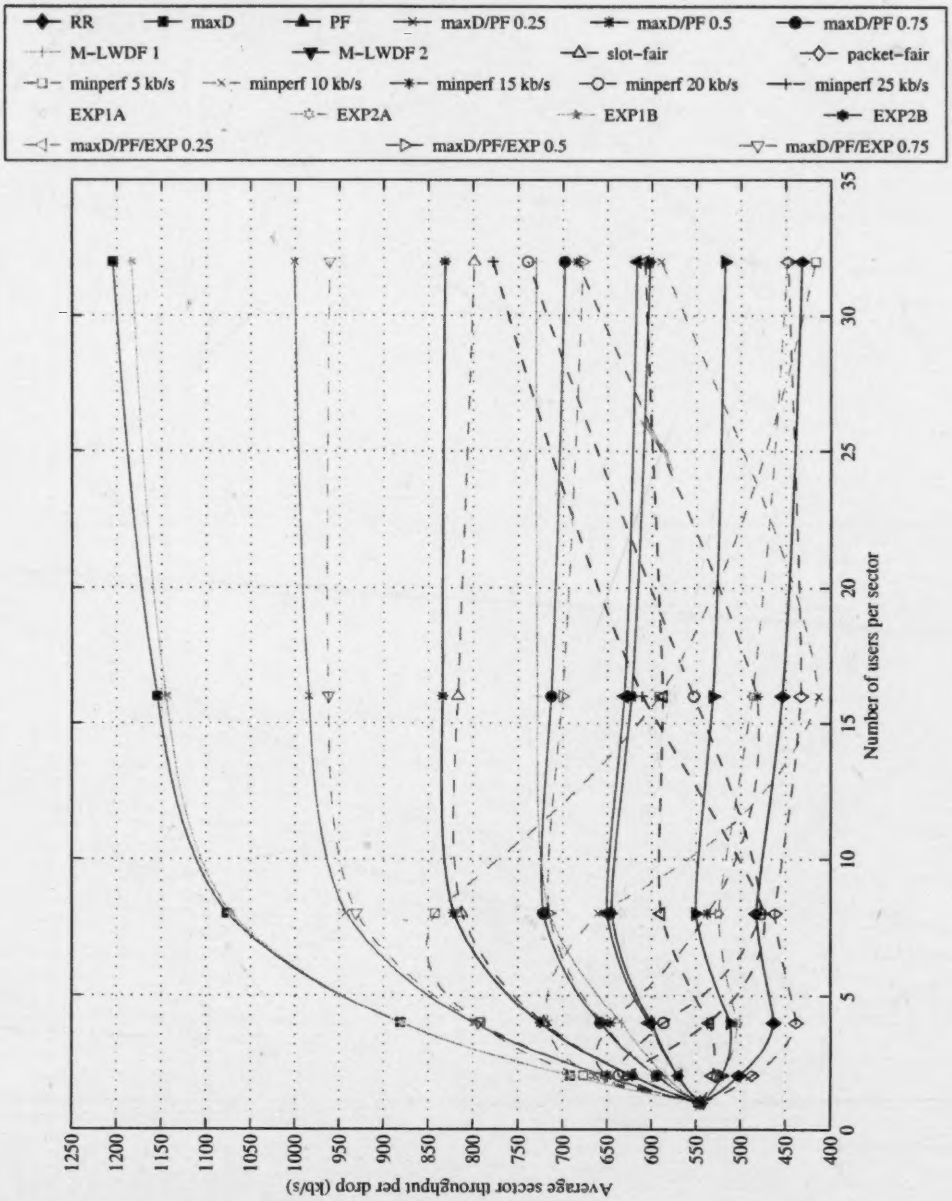


Figure E.1: Average sector throughput per drop vs. number of users per sector in the ITU vehicular B channel using outdated SIR$_p$ values and margins

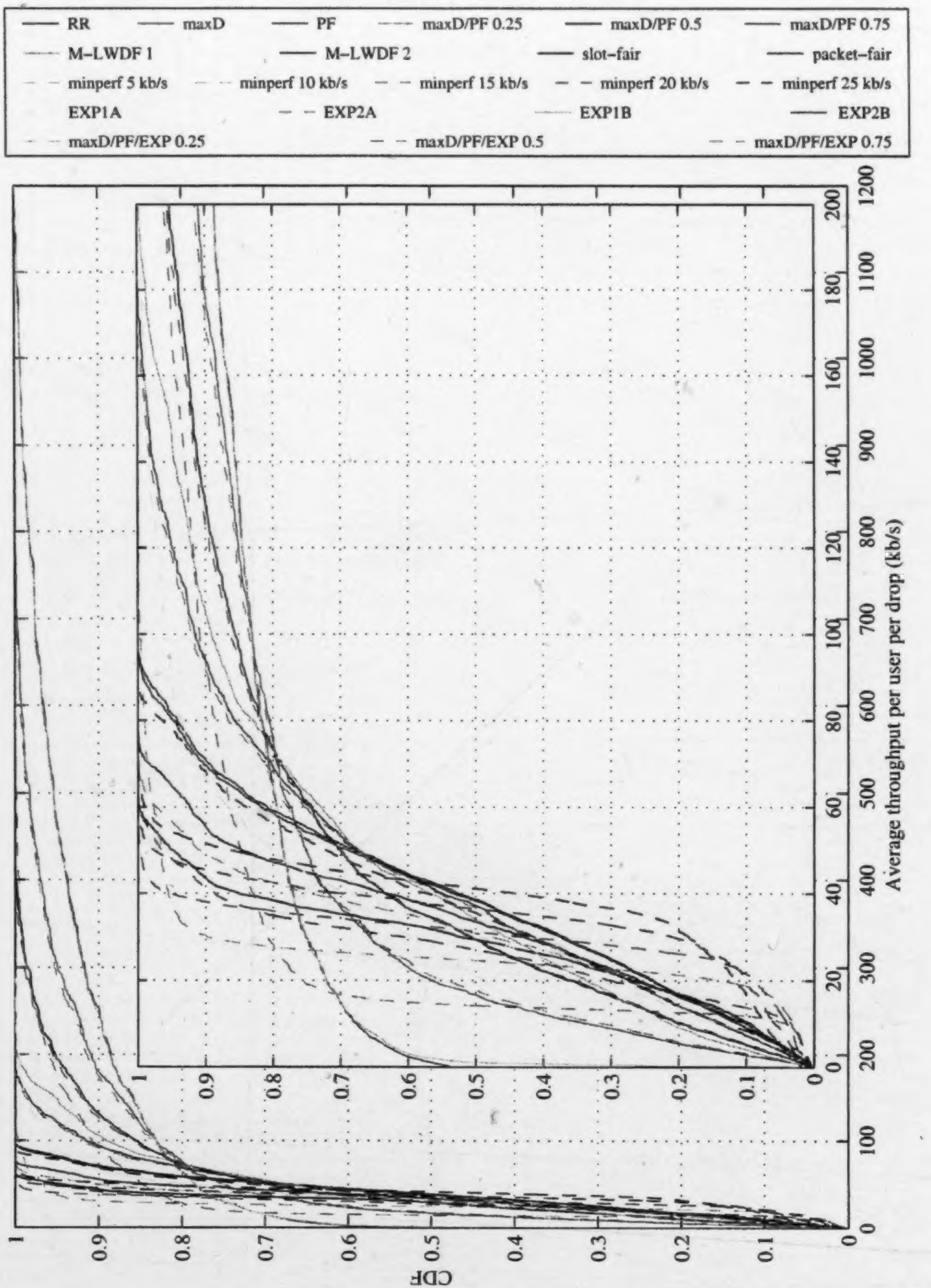


Figure E.2: Distributions of average throughput per user per drop in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins

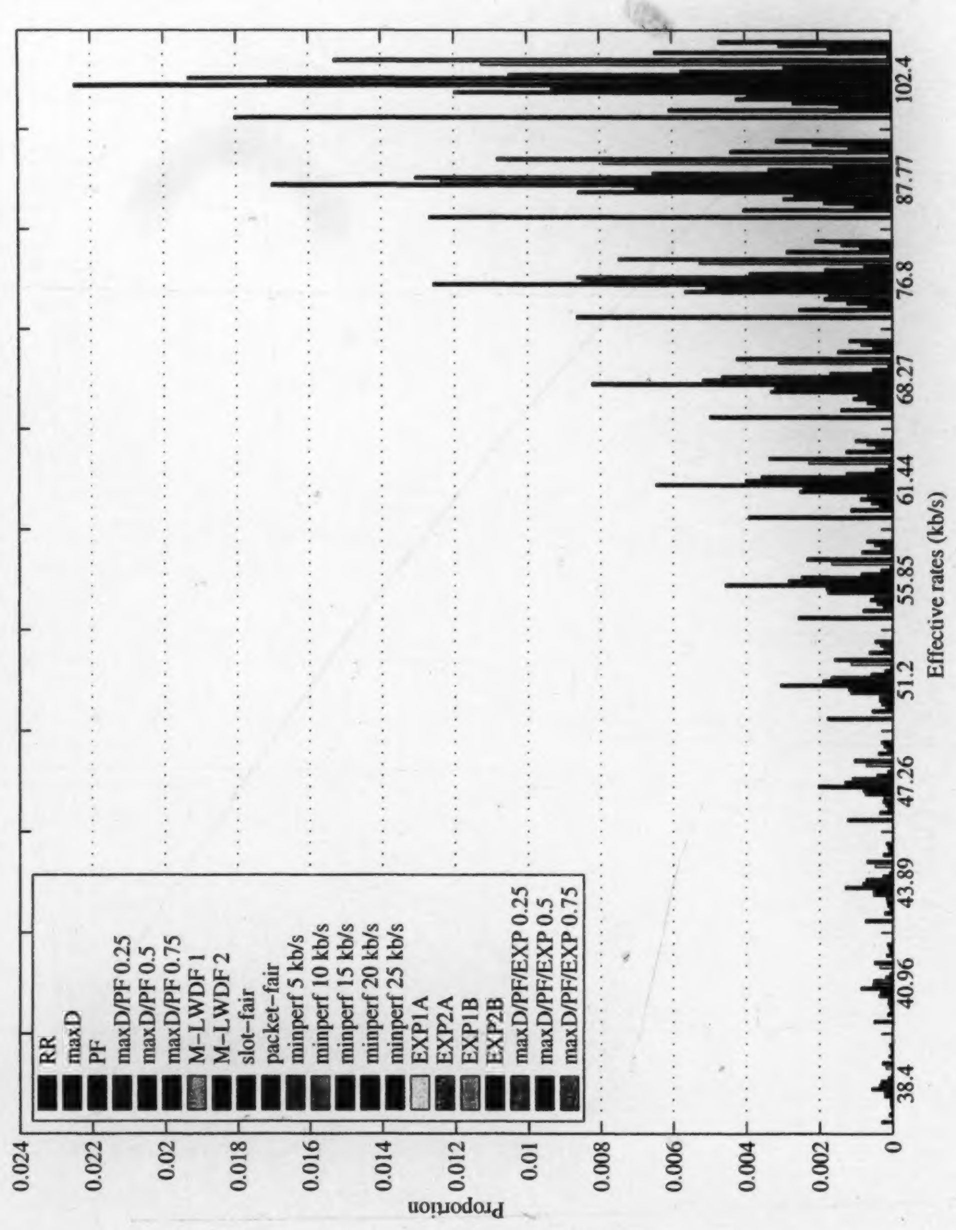


Figure E.3 (Part 1 of 2): Distribution of effective bit rates for transmitted packets in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins

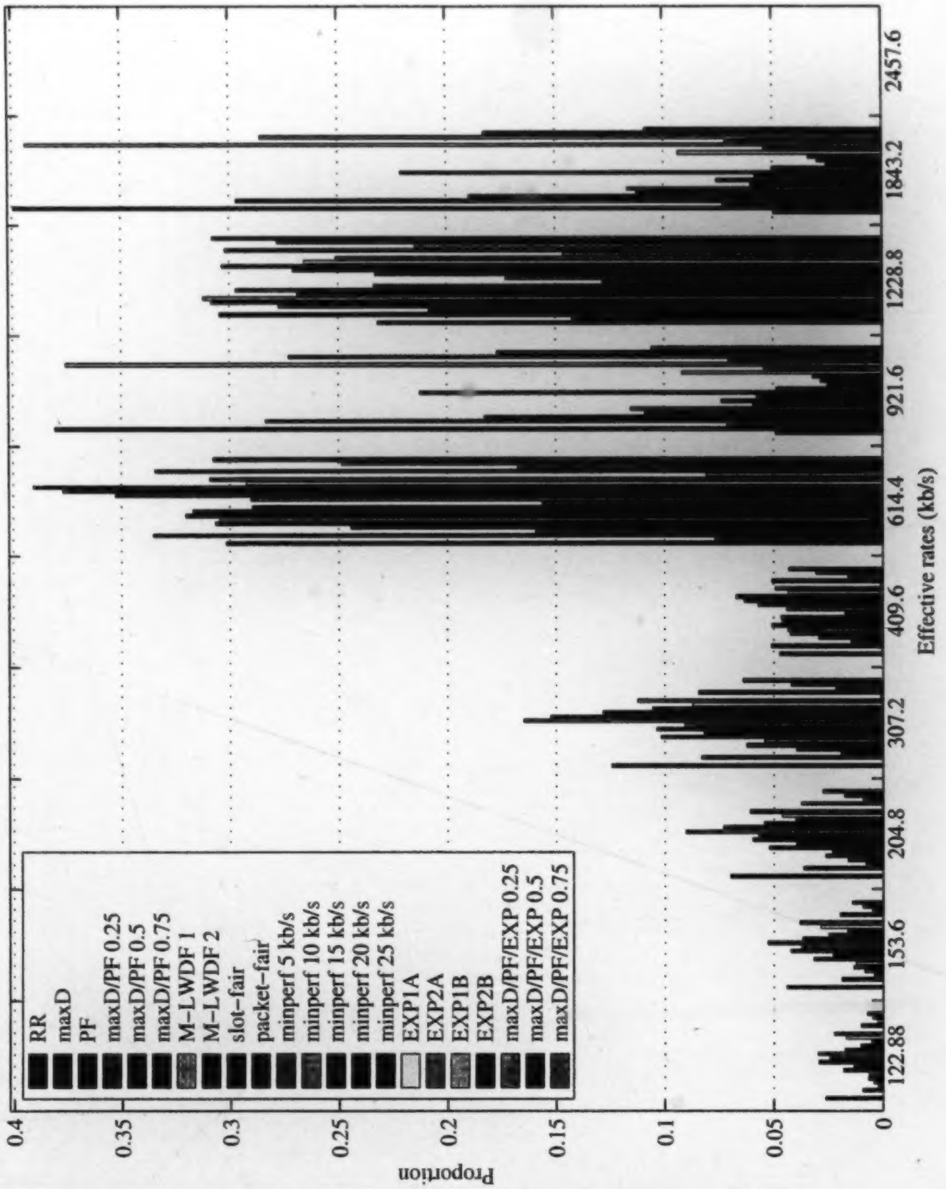
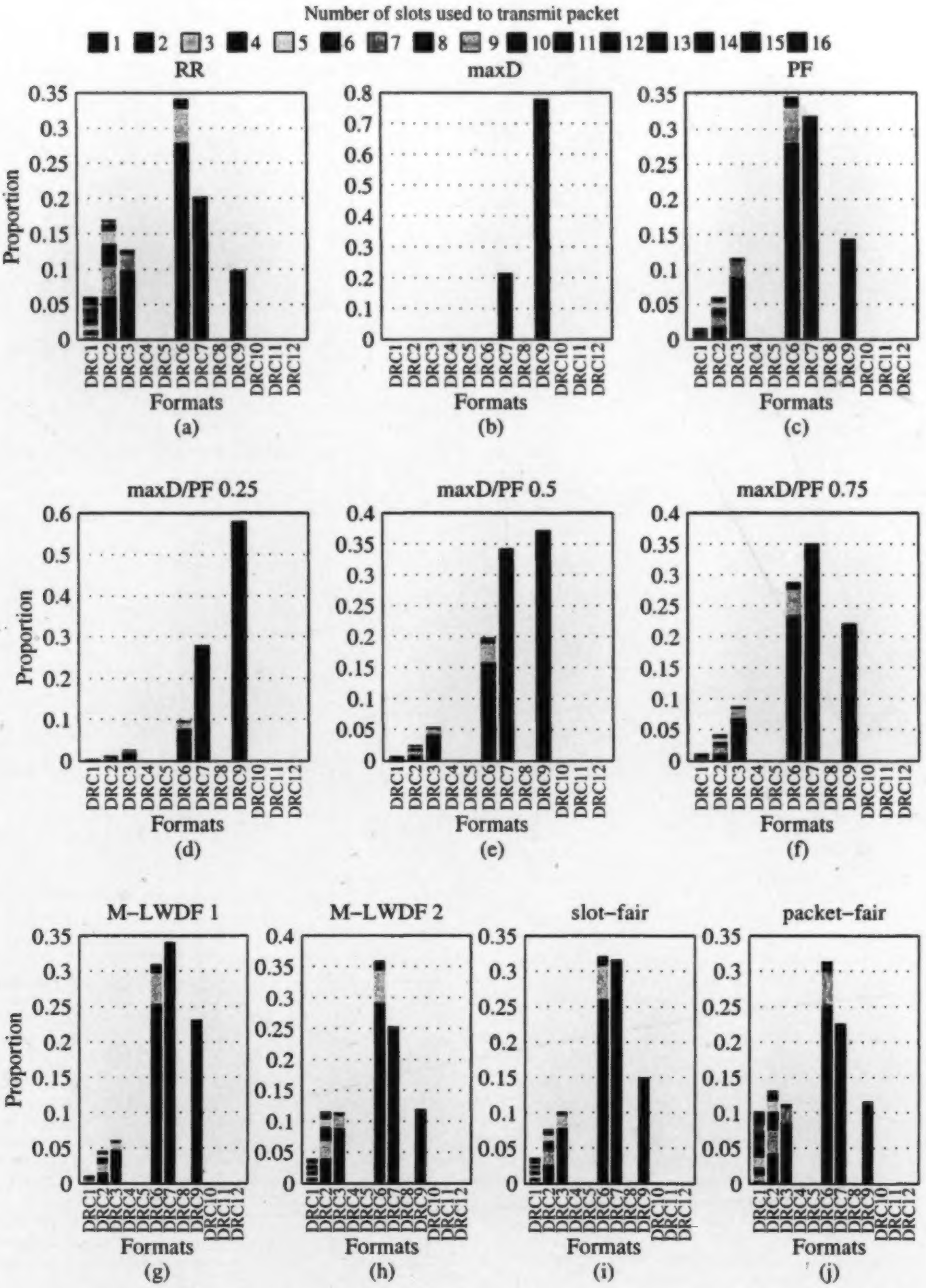


Figure E.3 (Part 2 of 2): Distribution of effective bit rates for transmitted packets in the ITU vehicular B channel with 16 users per sector using outdated SIR$_p$ values and margins

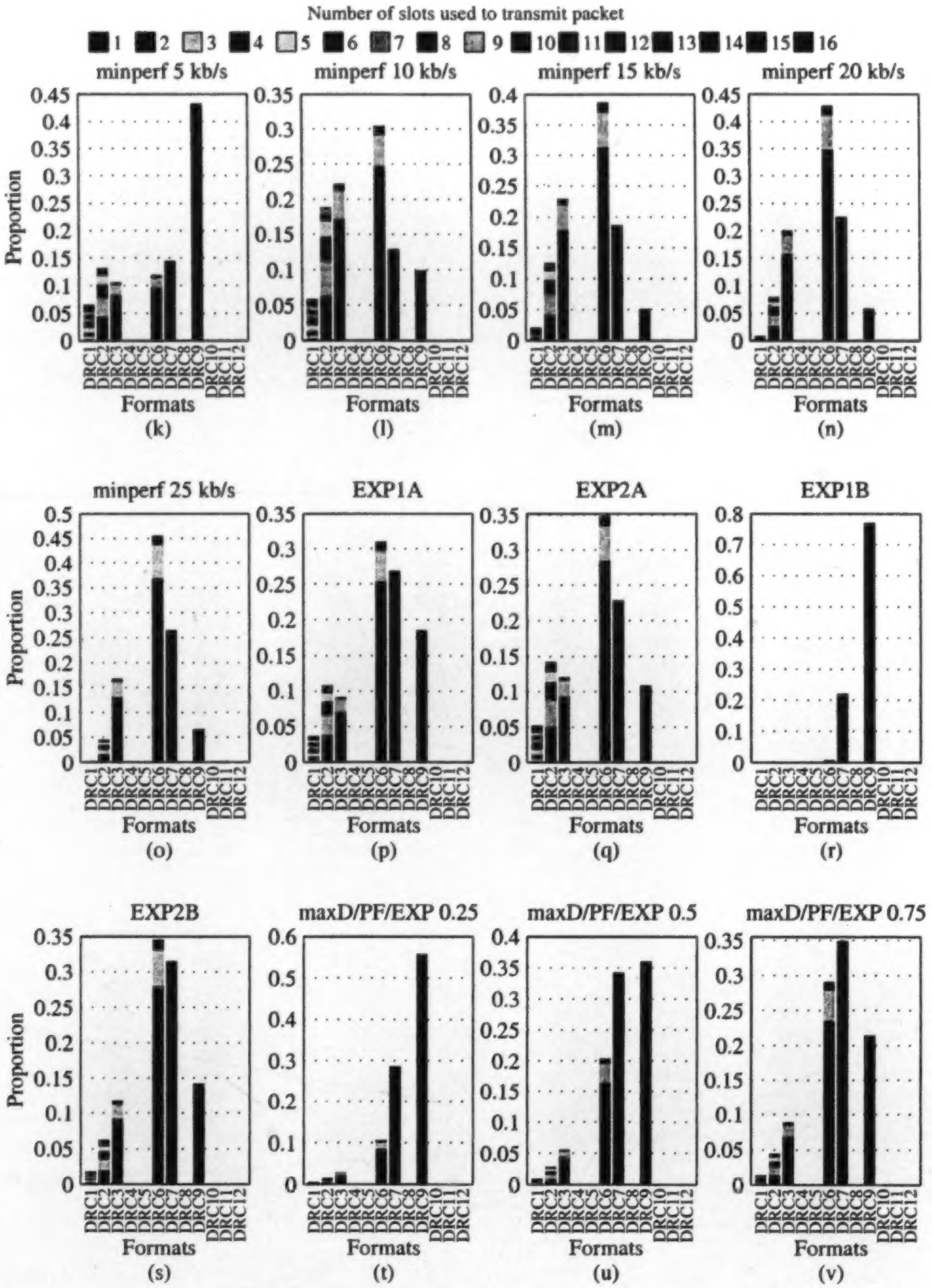


(a) (b) (c)

(d) (e) (f)

(g) (h) (i) (j)

Figure E.4 (Part 1 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins



(k) (l) (m) (n)

(o) (p) (q) (r)

(s) (t) (u) (v)

Figure E.4 (Part 2 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins

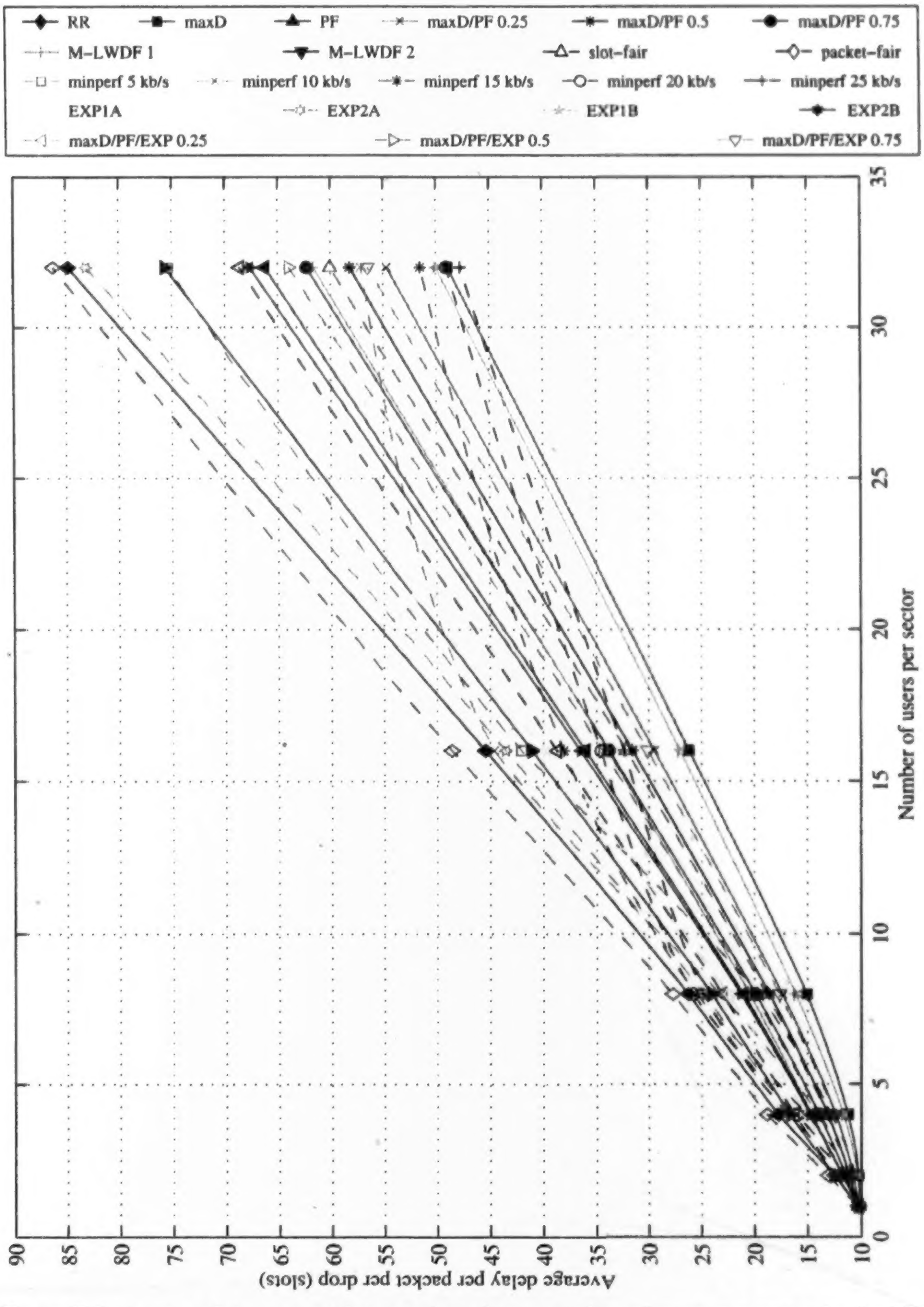


Figure E.5: Average delay per packet per drop versus number of users per sector in the ITU vehicular B channel using outdated SIR_p values and margins

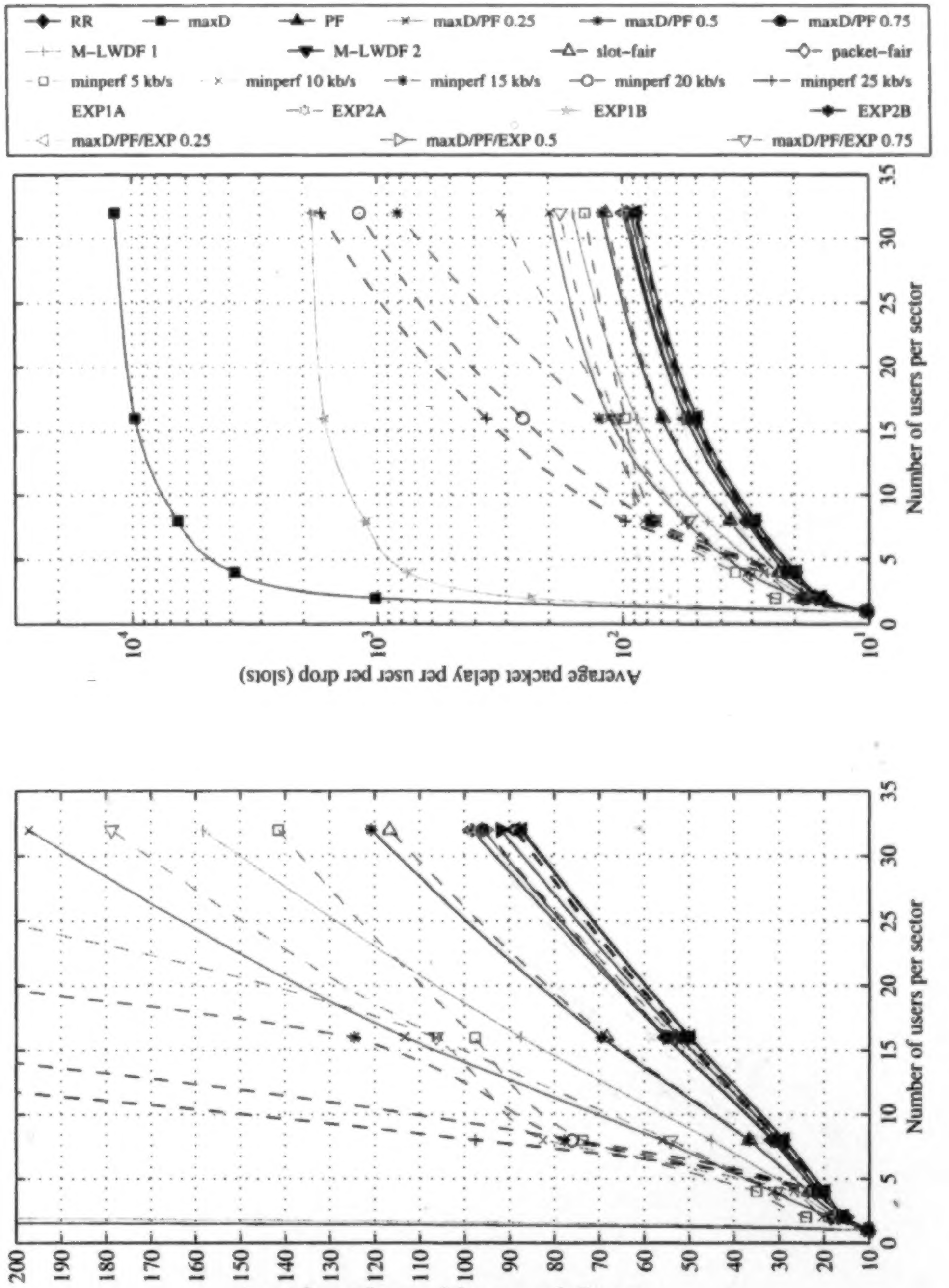


Figure E.6: Average packet delay per user per drop versus number of users per sector in the ITU vehicular B channel on a linear and a logarithmic scale using outdated SIR_p values and margins

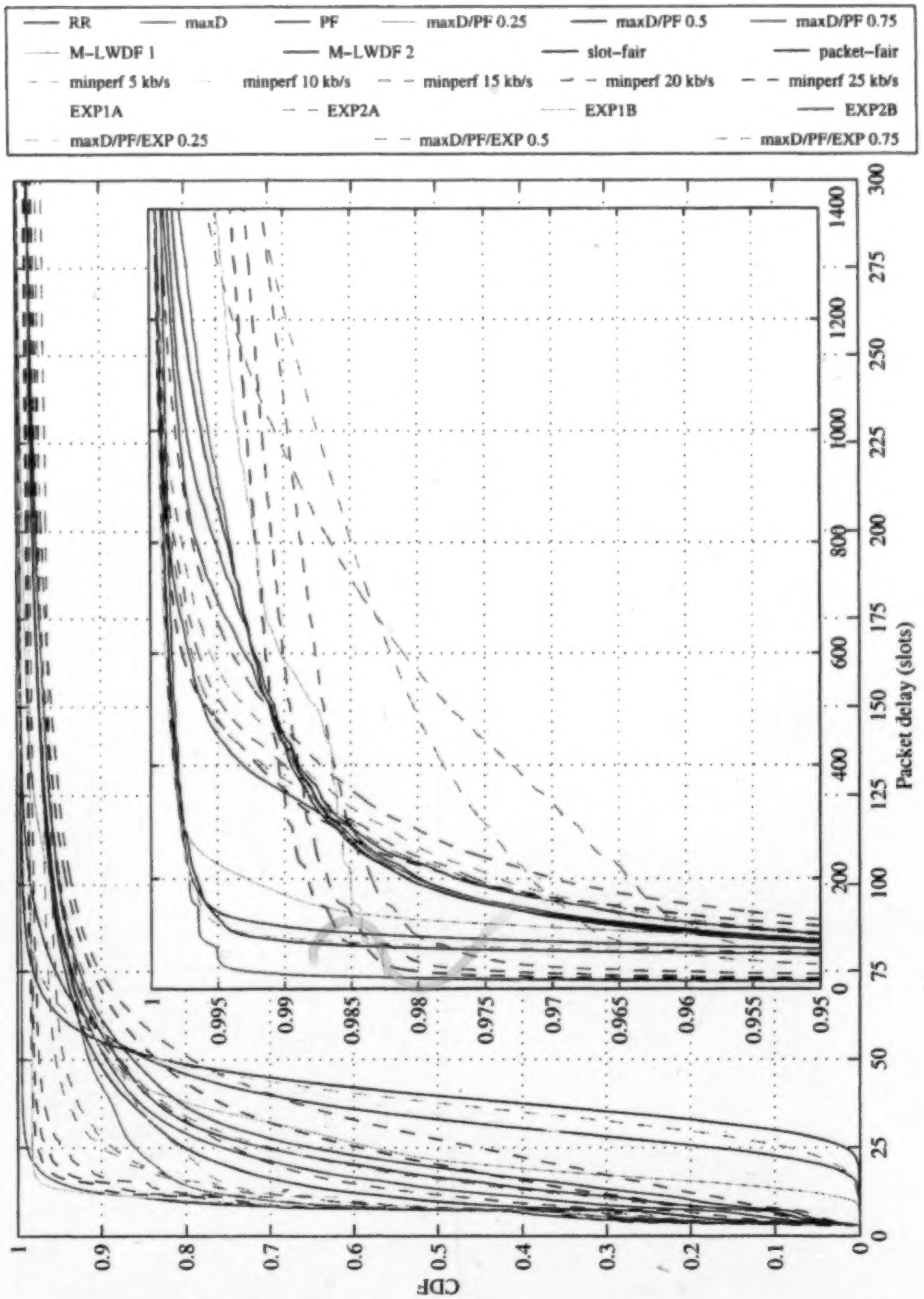


Figure E.7: Cumulative distribution functions of delays experienced by packets in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins

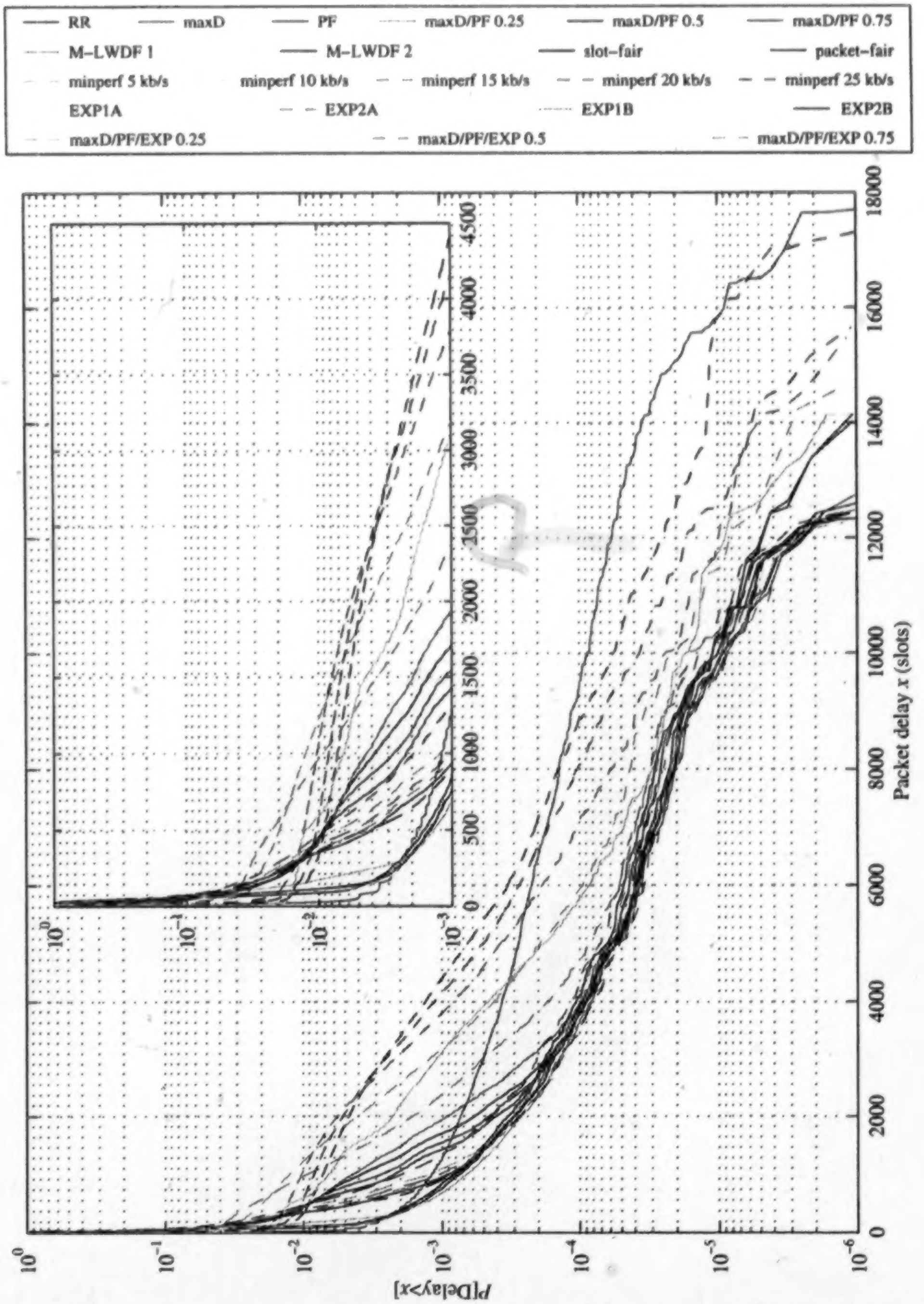


Figure E.8: Probability of delays experienced by packets being larger than a given value in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins

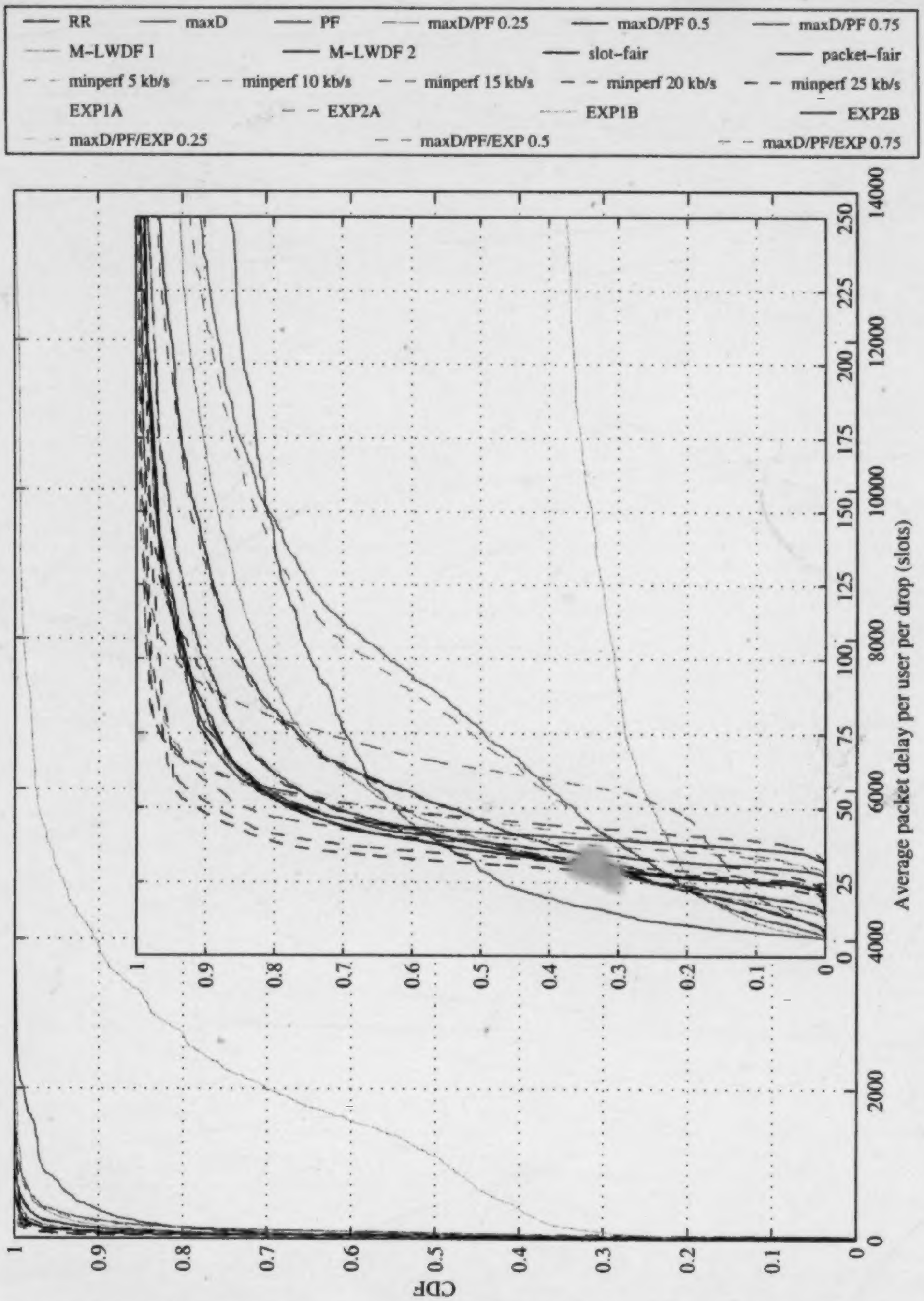


Figure E.9: Distributions of average packet delay per user for users who receive at least one packet in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins

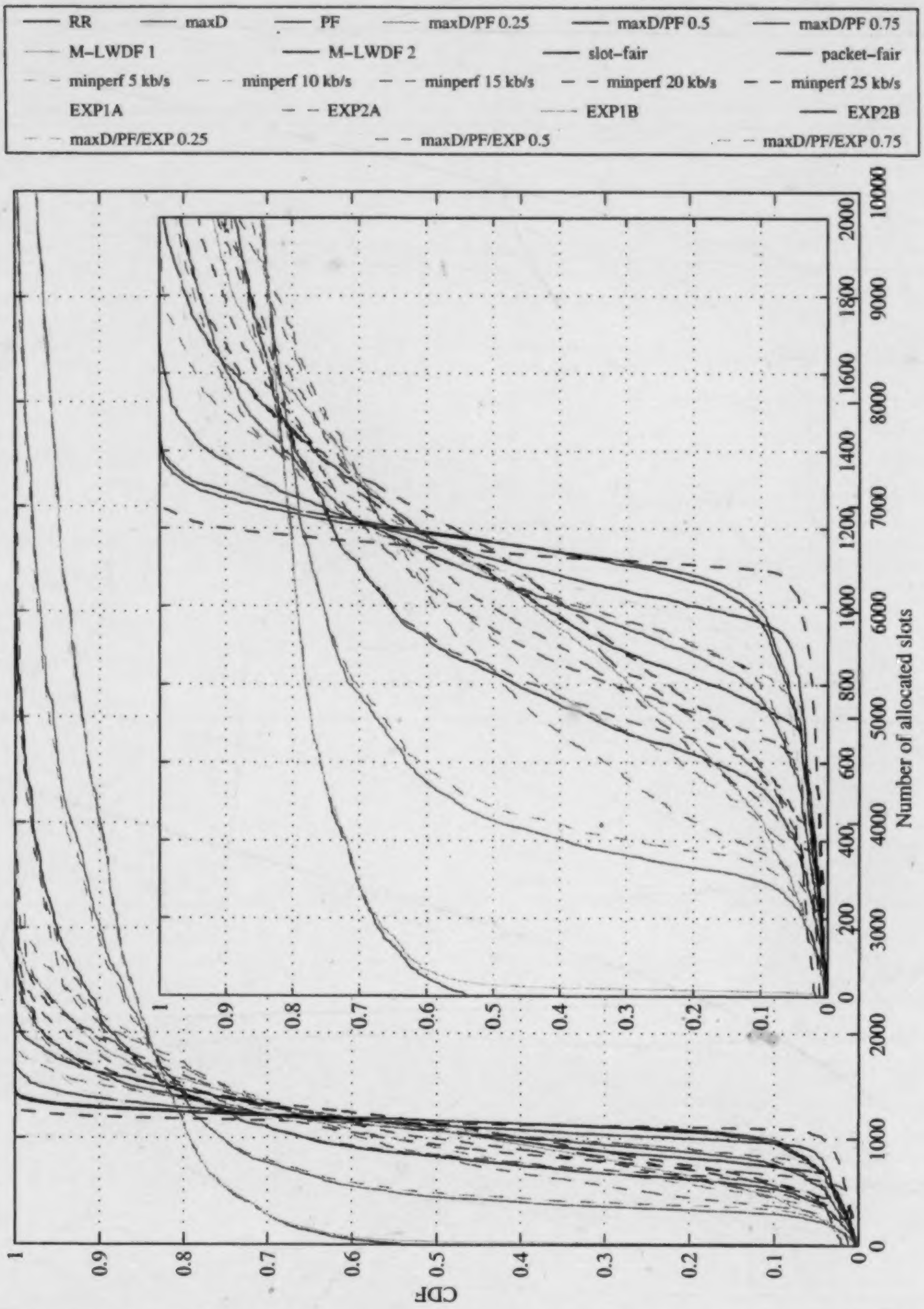


Figure E.10: Distributions of allocated slots per user per drop in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins

Appendix F: ITU Vehicular B Channel Results Comparing the "PF A" and "PF B" Average Rate Calculation Methods using Outdated SIR$_p$ Values and Margins

This appendix contains graphs comparing the simulation results using the "PF A" and "PF B" methods of calculating the average rate in the metrics of scheduling algorithms for the ITU vehicular B channel when outdated SIR$_p$ values (i.e. the signal-to-interference ratio minus the Rayleigh component) plus margins were used to determine the transmission format each user could support.

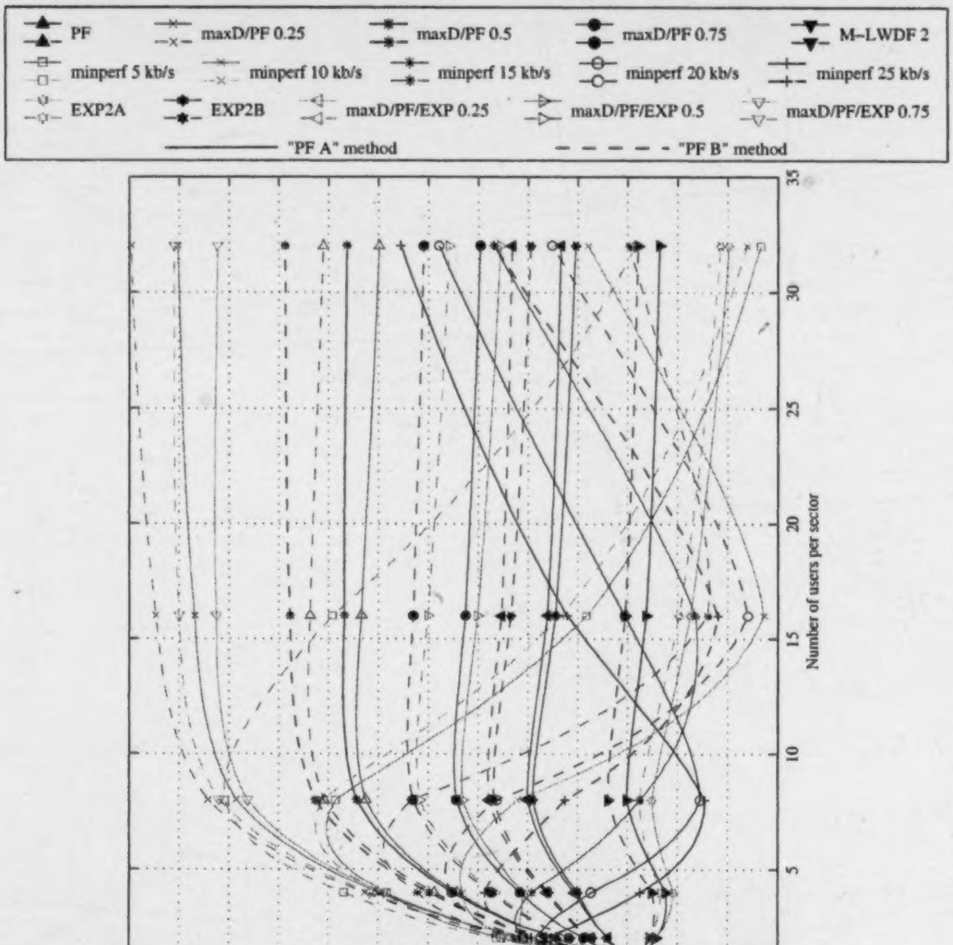


Figure F.1: Average sector throughput per drop vs. number of users per sector in the ITU vehicular B channel using outdated SIR$_p$ values and margins and comparing the "PF A" and "PF B" average rate calculation methods

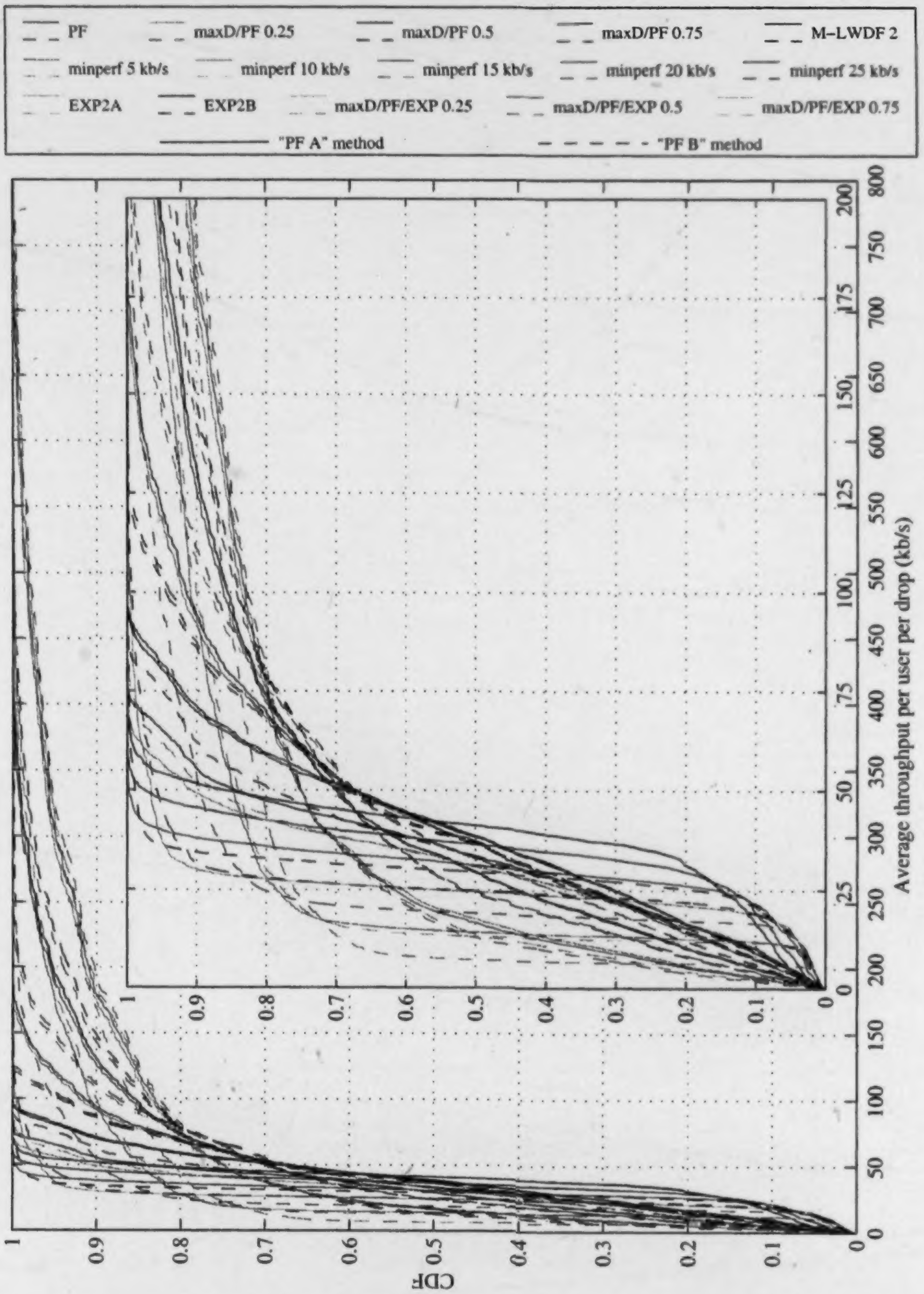


Figure F.2: Distributions of average throughput per user per drop in the ITU vehicular B channel with 16 users per sector using outdated $\mathrm{SIR_p}$ values and margins and comparing the "PF A" and "PF B" average rate calculation methods

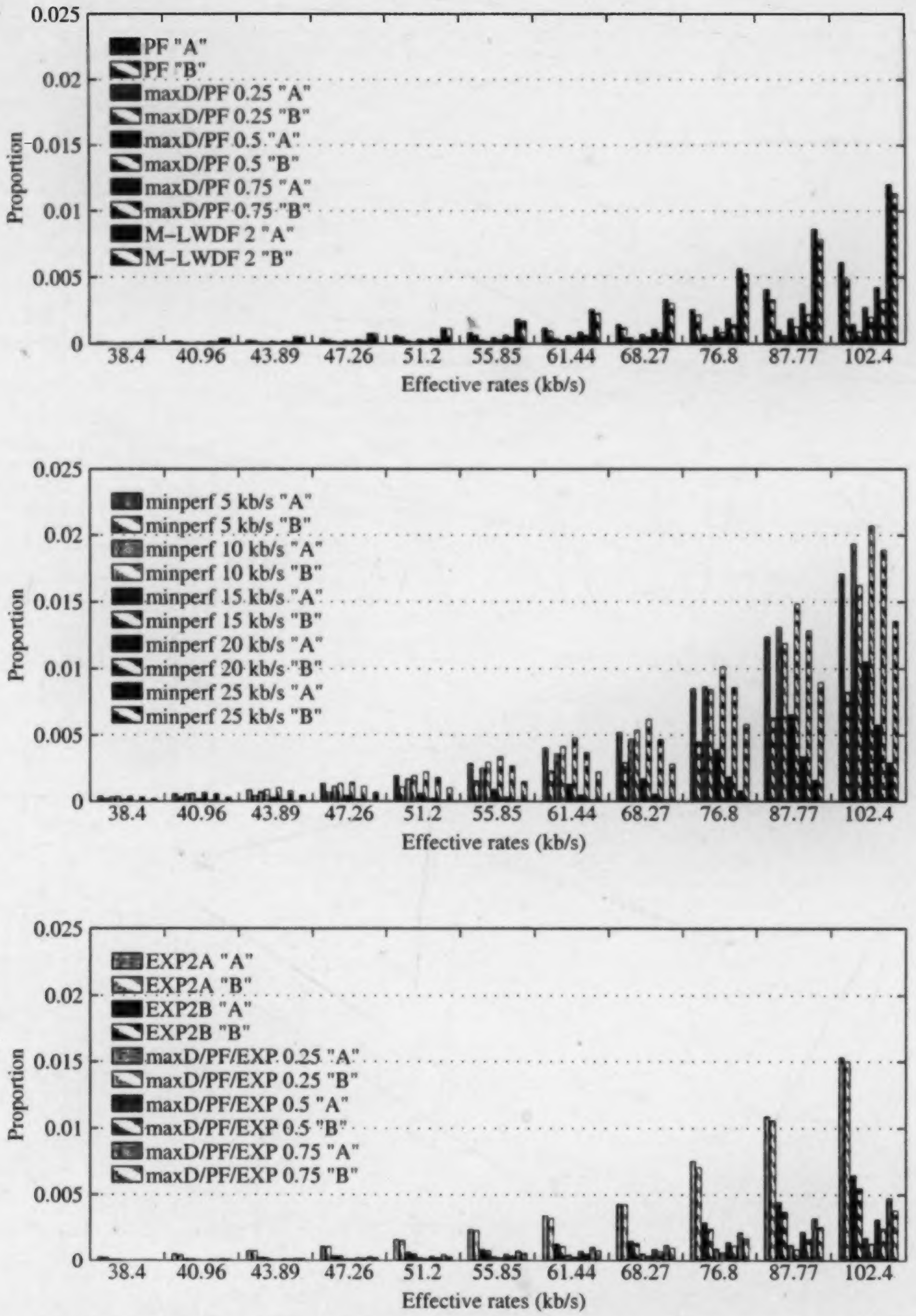


Figure F.3 (Part 1 of 2): Distribution of effective bit rates for transmitted packets in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins and comparing the “PF A” and “PF B” average rate calculation methods

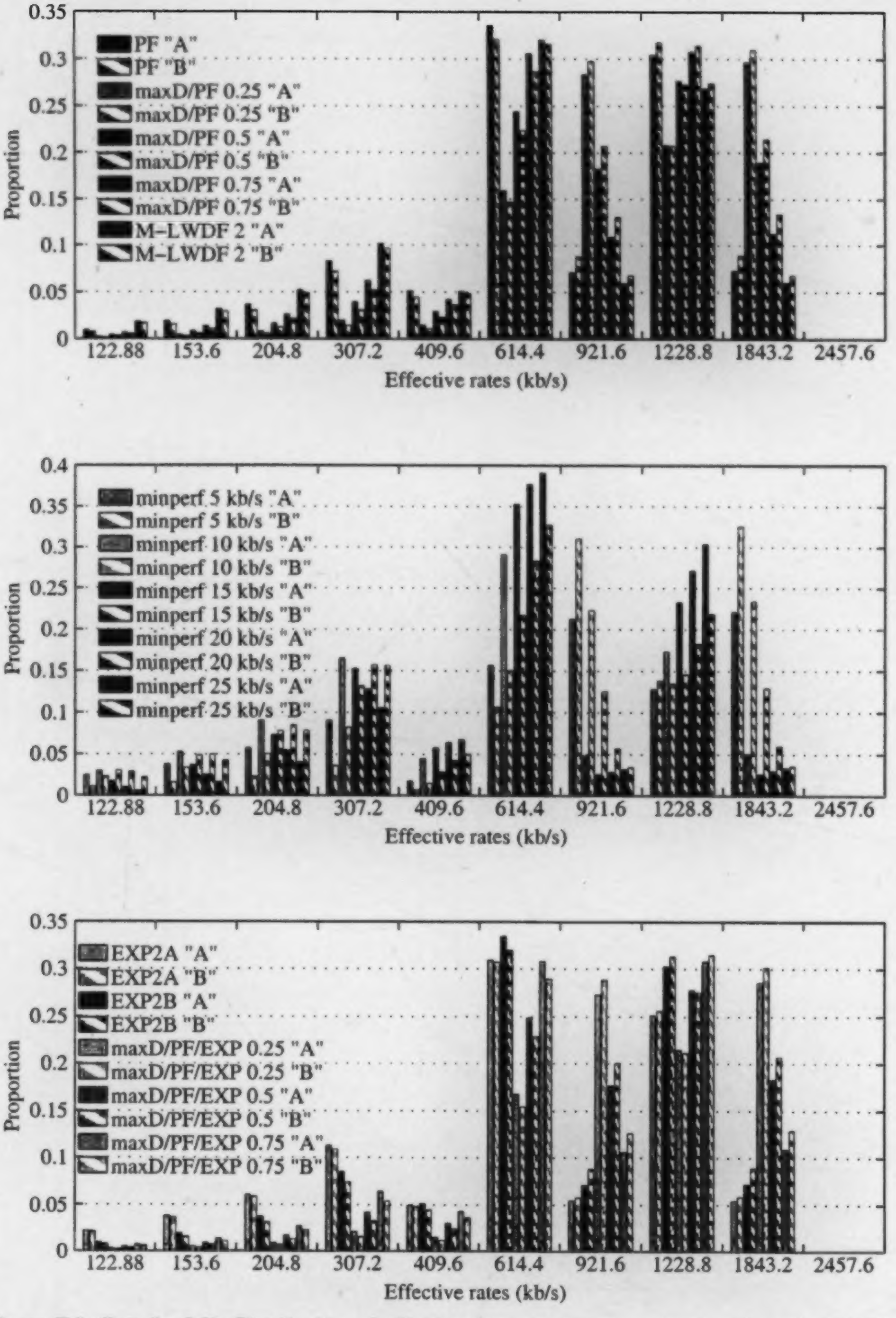
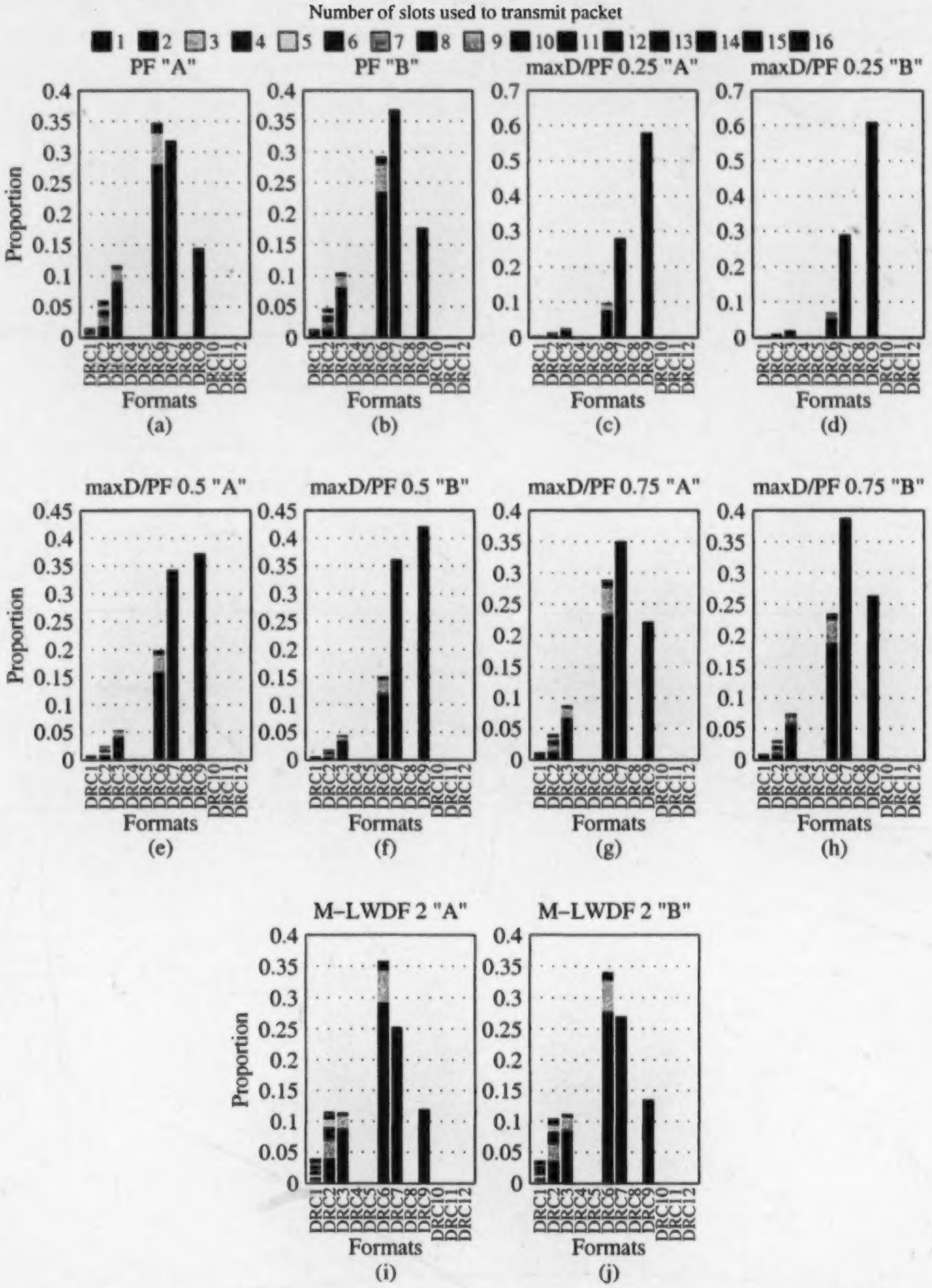
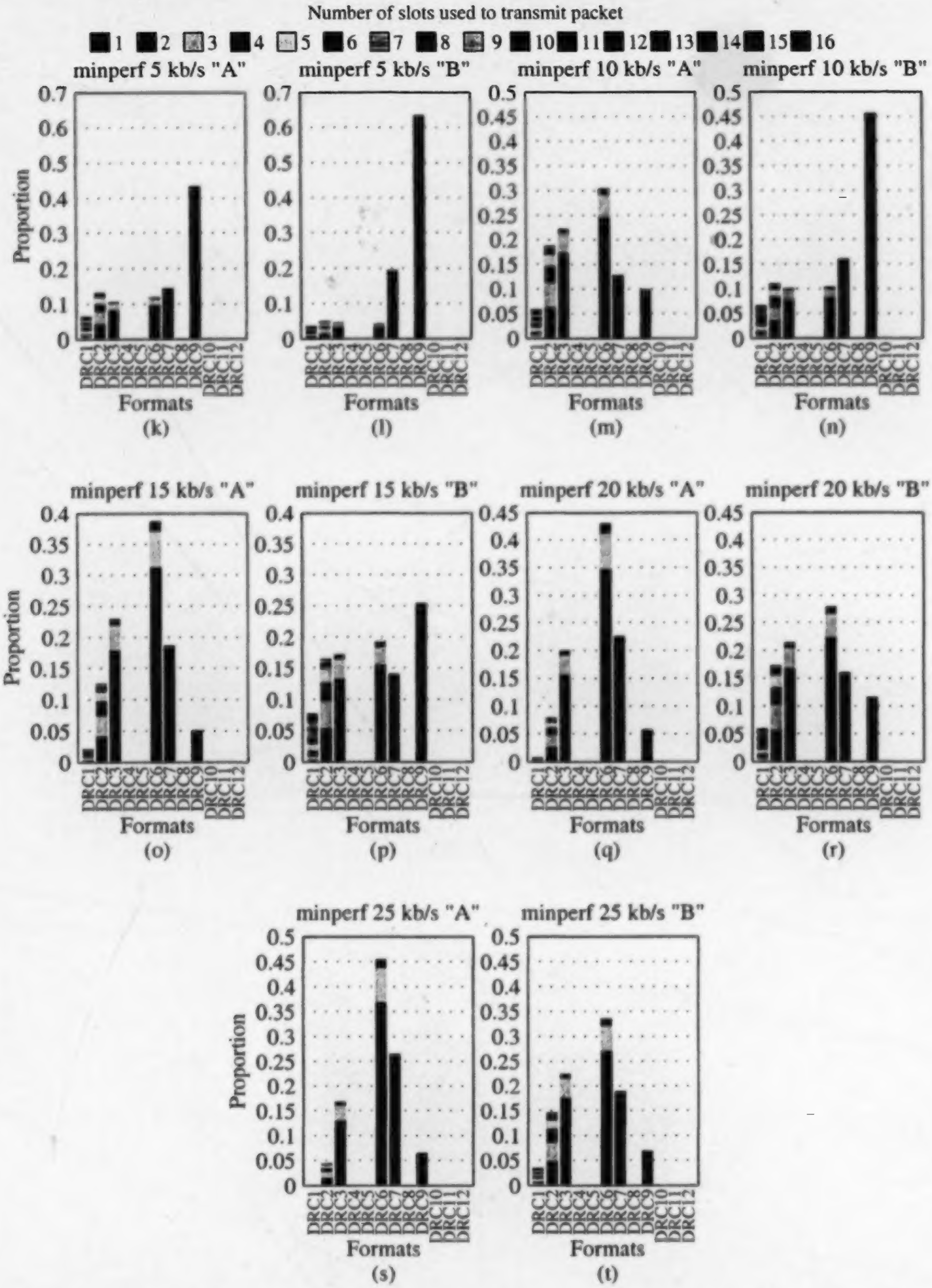


Figure F.3 (Part 2 of 2): Distribution of effective bit rates for transmitted packets in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins and comparing the “PF A” and “PF B” average rate calculation methods



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure F.4 (Part 1 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins and comparing the "PF A" and "PF B" average rate calculation methods



(k) (l) (m) (n)

(o) (p) (q) (r)

(s) (t)

Figure F.4 (Part 2 of 3): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins and comparing the "PF A" and "PF B" average rate calculation methods

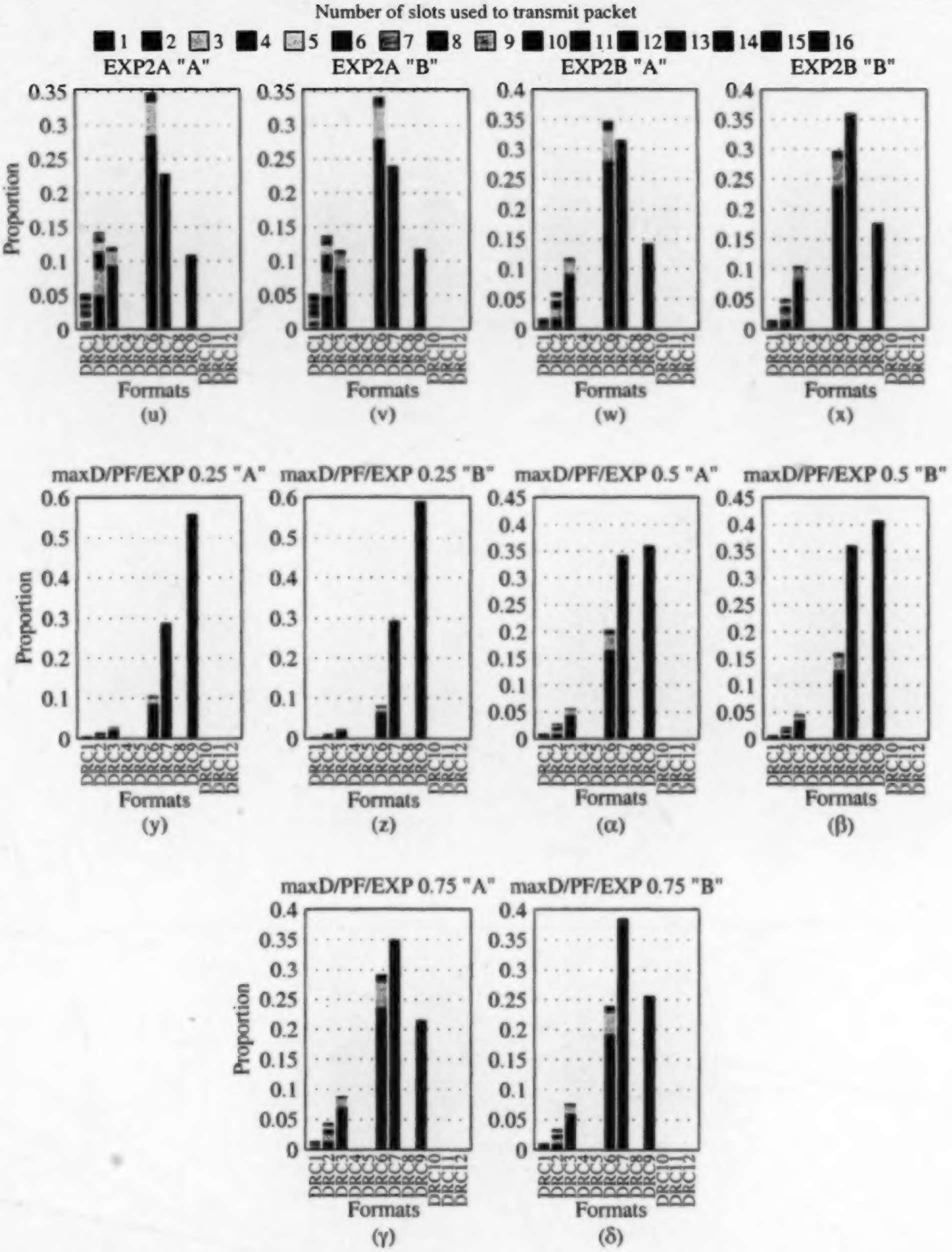


Figure F.4 (Part 3 of 3): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins and comparing the "PF A" and "PF B" average rate calculation methods

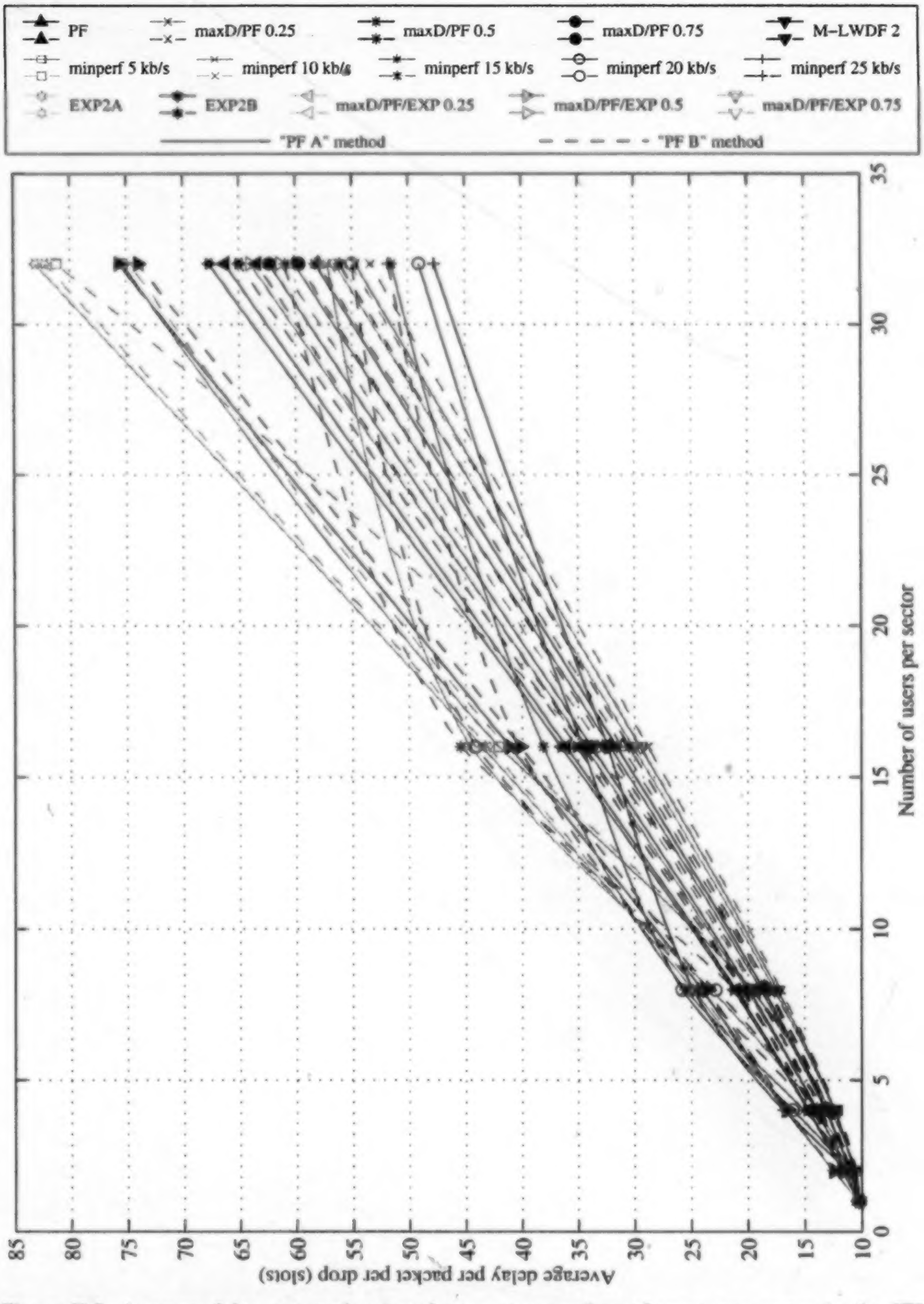


Figure F.5: Average delay per packet per drop versus number of users per sector in the ITU vehicular B channel using outdated SIR_p values and margins and comparing the “PF A” and “PF B” average rate calculation methods

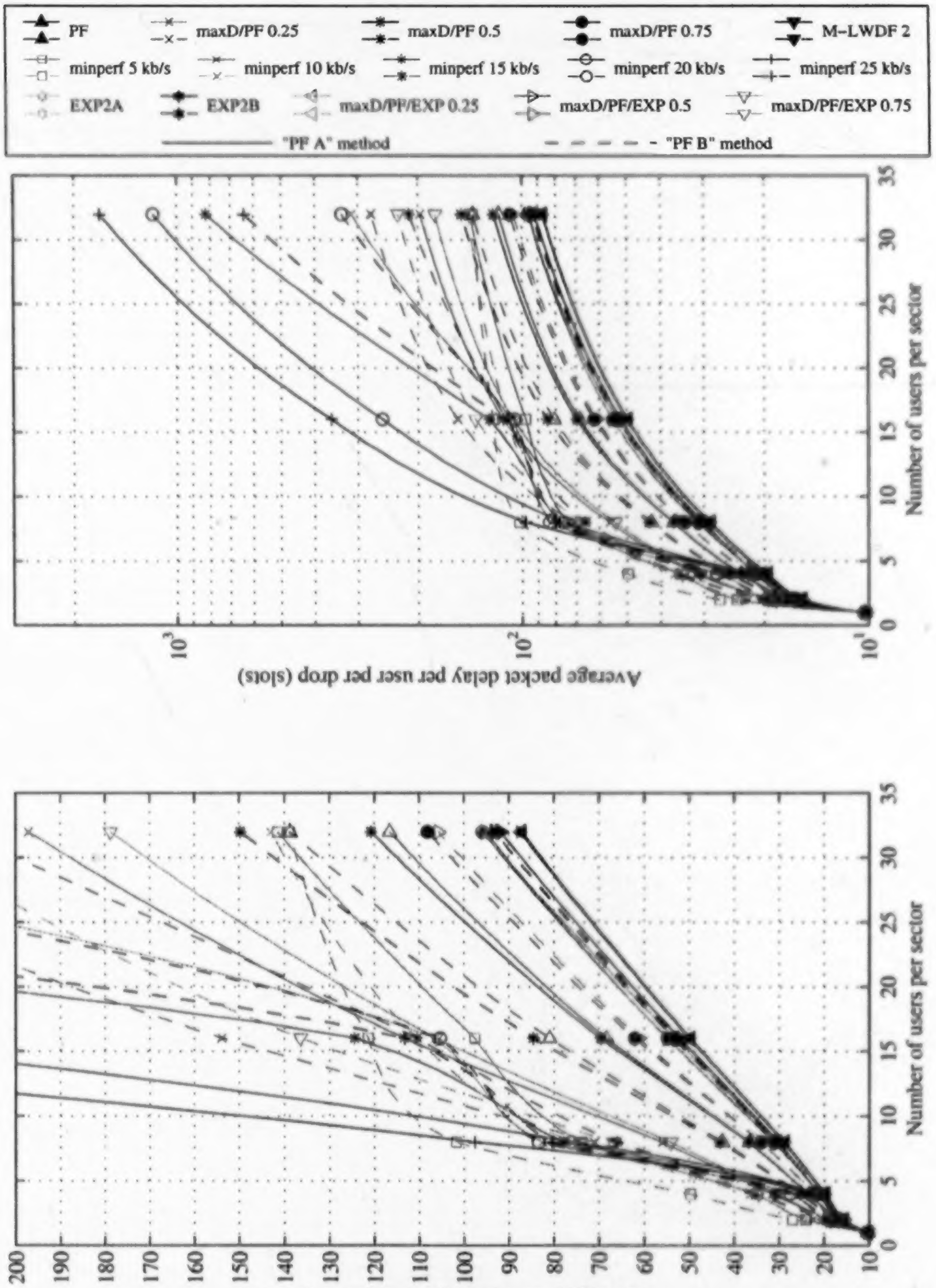


Figure F.6: Average packet delay per user per drop versus number of users per sector in the ITU vehicular B channel on a linear and a logarithmic scale using outdated SIR_p values and margins and comparing the "PF A" and "PF B" average rate calculation methods

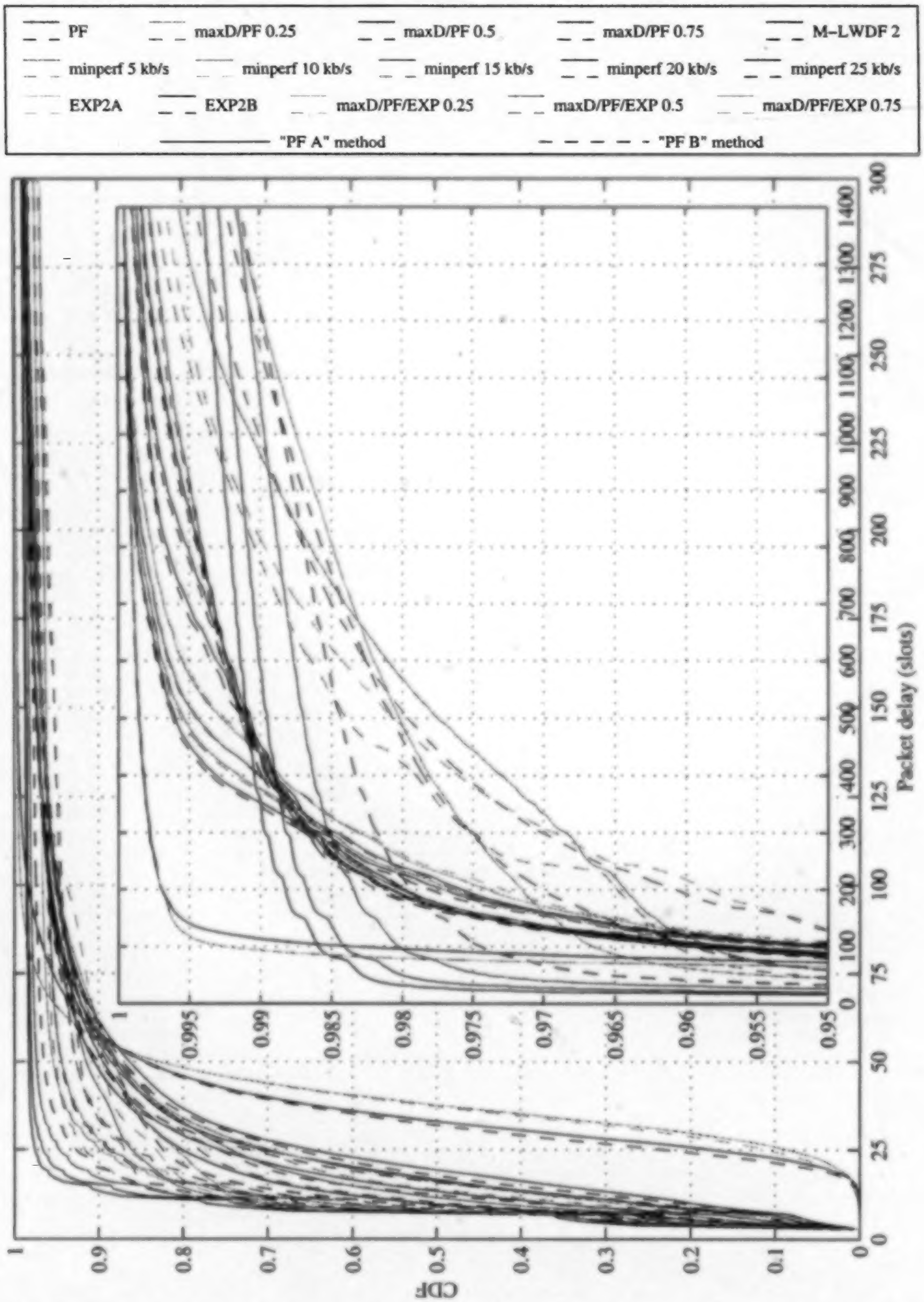


Figure F.7: Cumulative distribution functions of delays experienced by packets in the ITU vehicular B channel with 16 users per sector using outdated SIR$_p$ values and margins and comparing the "PF A" and "PF B" average rate calculation methods

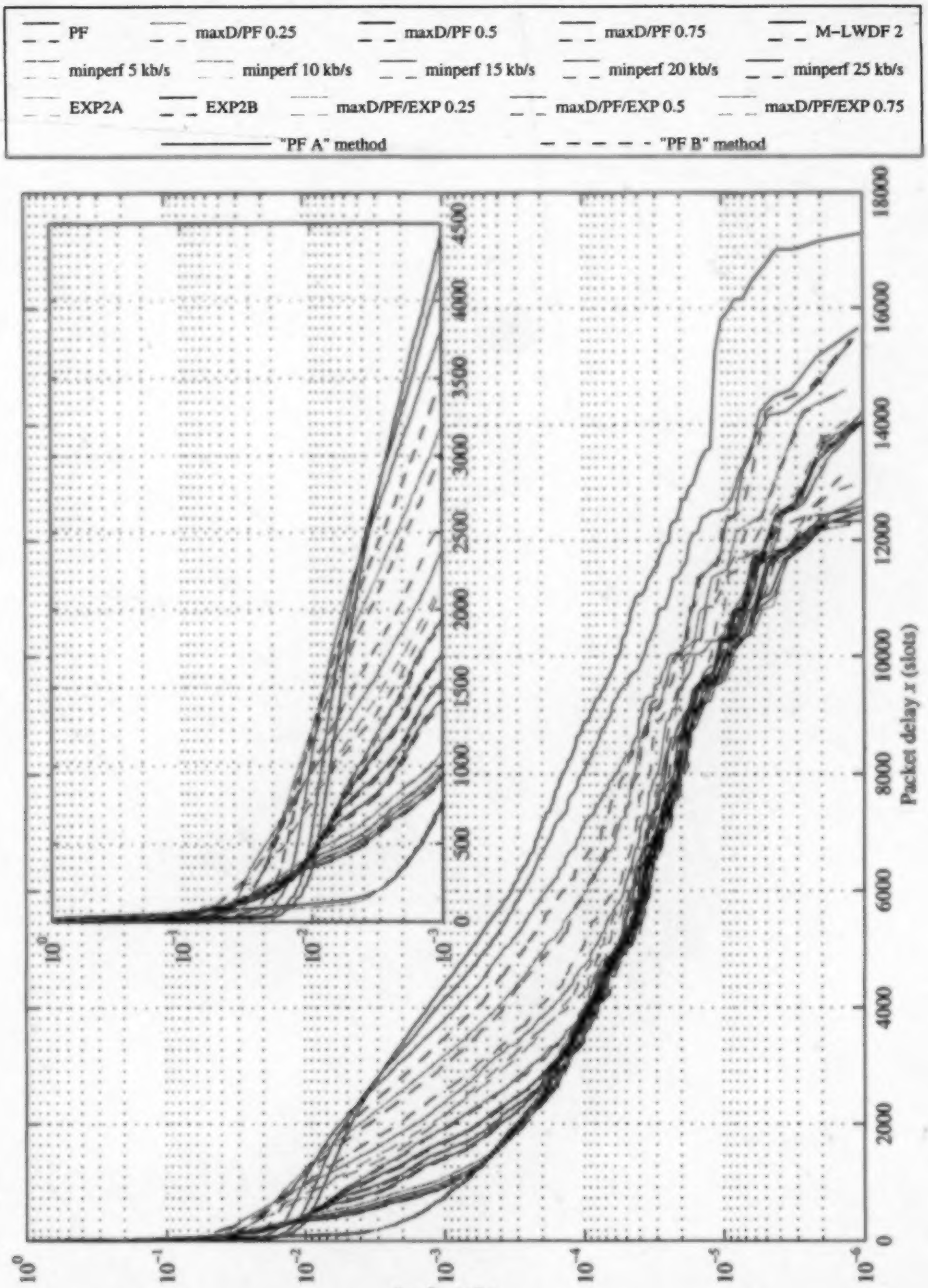


Figure F.8: Probability of delays experienced by packets being larger than a given value in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins and comparing the "PF A" and "PF B" average rate calculation methods

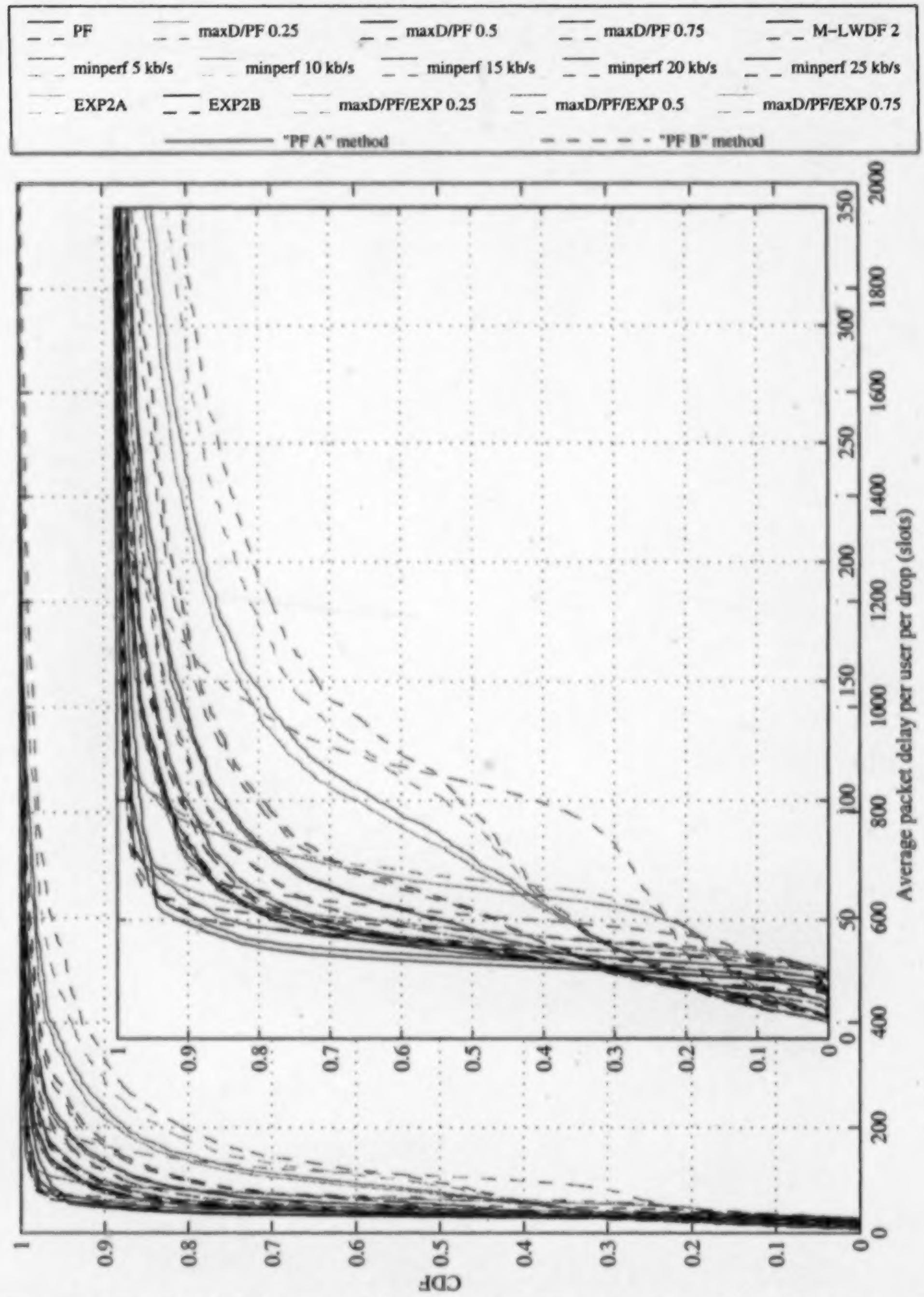


Figure F.9: Distributions of average packet delay per user for users who receive at least one packet in the ITU vehicular B channel with 16 users per sector using outdated SIR$_p$ values and margins and comparing the "PF A" and "PF B" average rate calculation methods

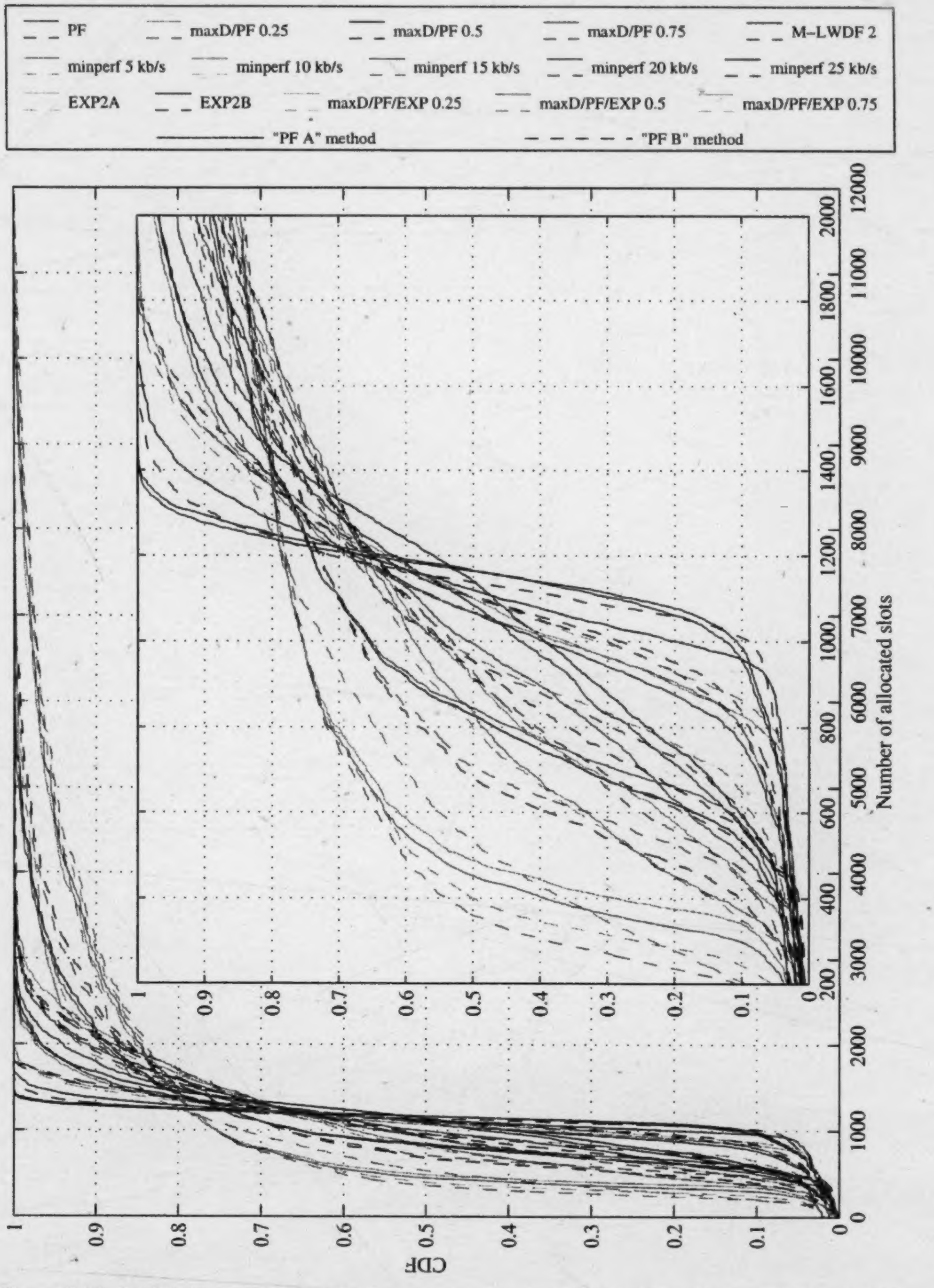


Figure F.10: Distributions of allocated slots per user per drop in the ITU vehicular B channel with 16 users per sector using outdated SIR_p values and margins and comparing the “PF A” and “PF B” average rate calculation methods

Appendix G: ITU Vehicular B Channel Results using Perfect One-slot Prediction and Margins

This appendix contains graphs showing the simulation results for the ITU vehicular B channel when perfect one-slot prediction plus margins was used to determine the transmission format each user could support.

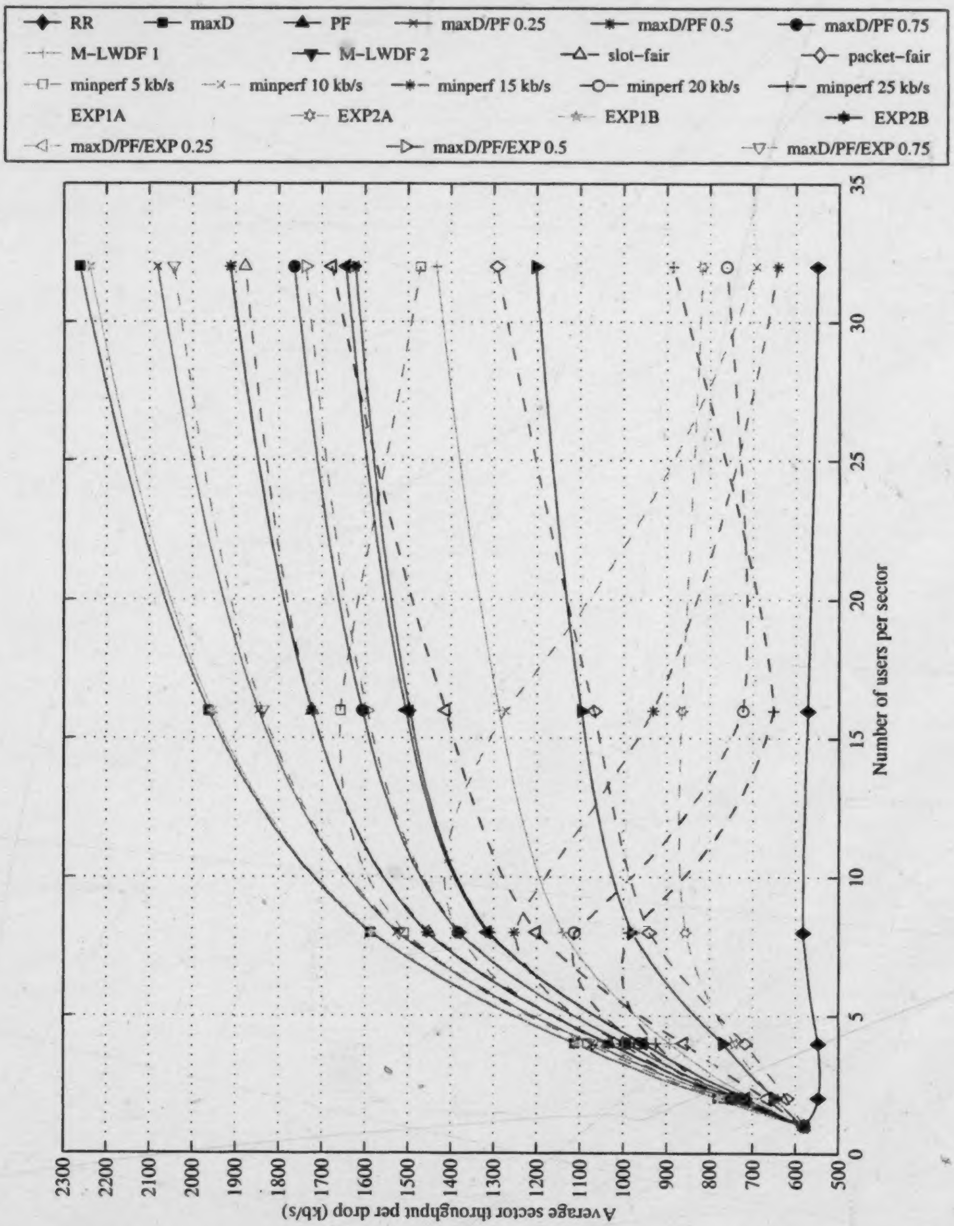


Figure G.1: Average sector throughput per drop vs. number of users per sector in the ITU vehicular B channel using perfect one-slot prediction and margins

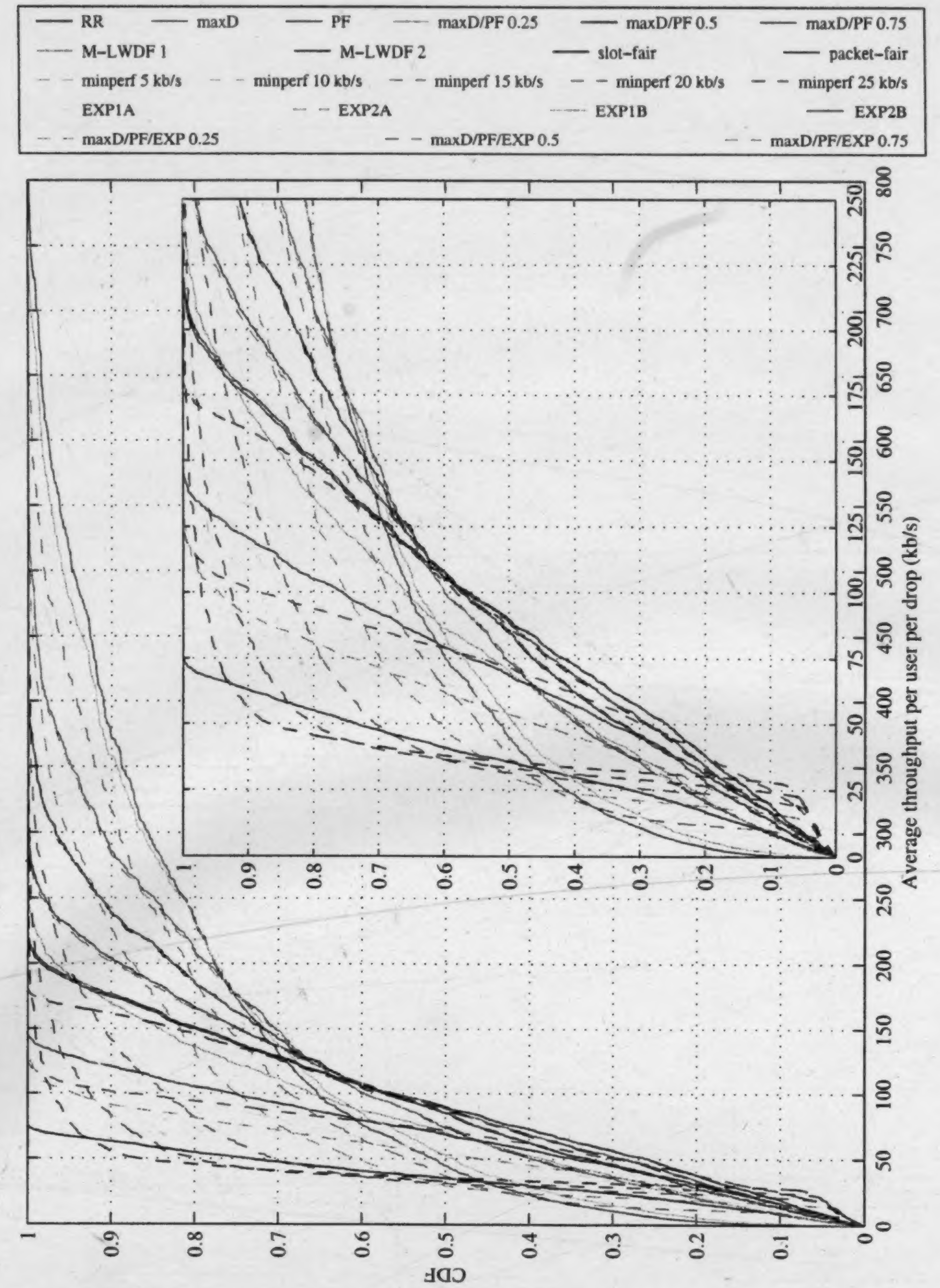


Figure G.2: Distributions of average throughput per user per drop in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins

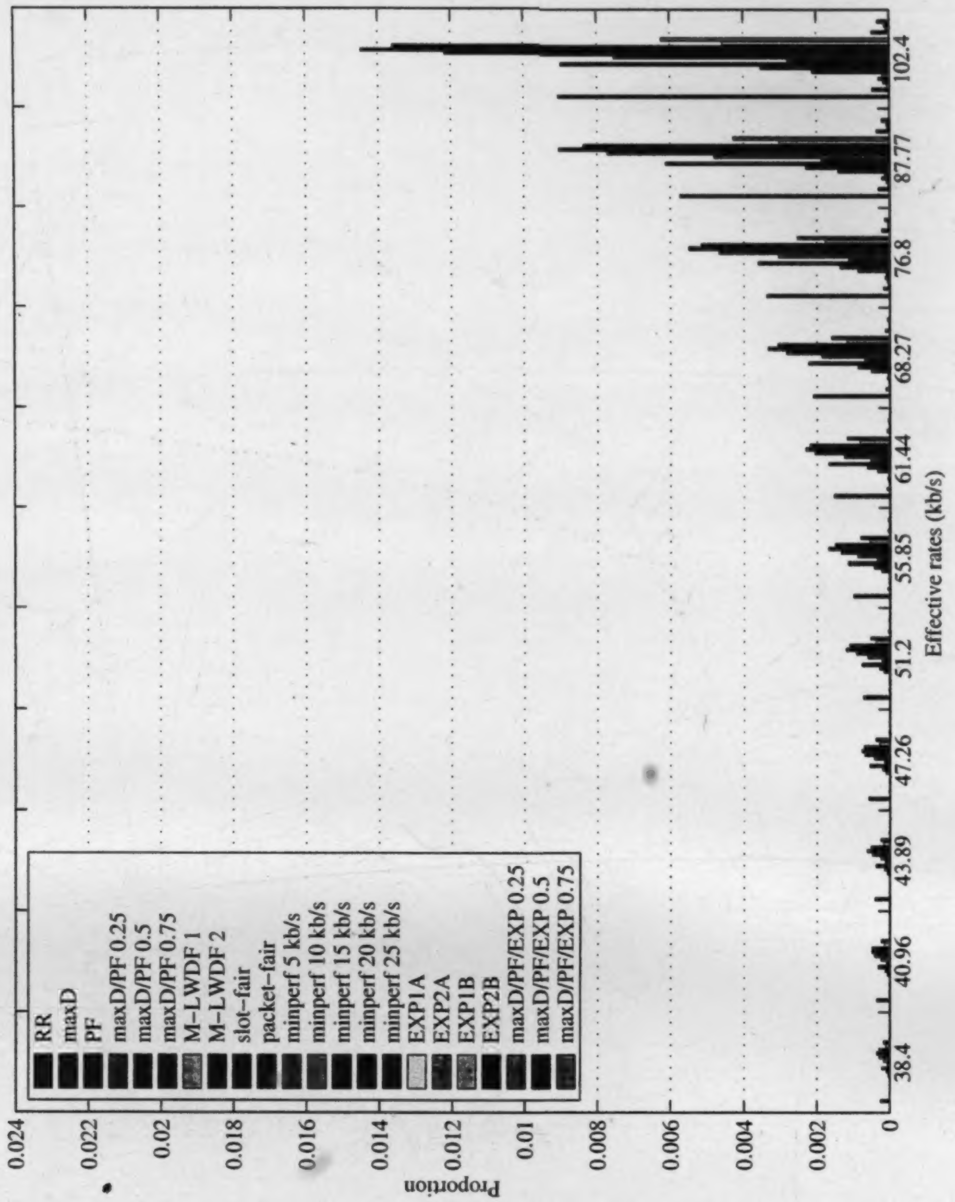


Figure G.3 (Part 1 of 2): Distribution of effective bit rates for transmitted packets in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins

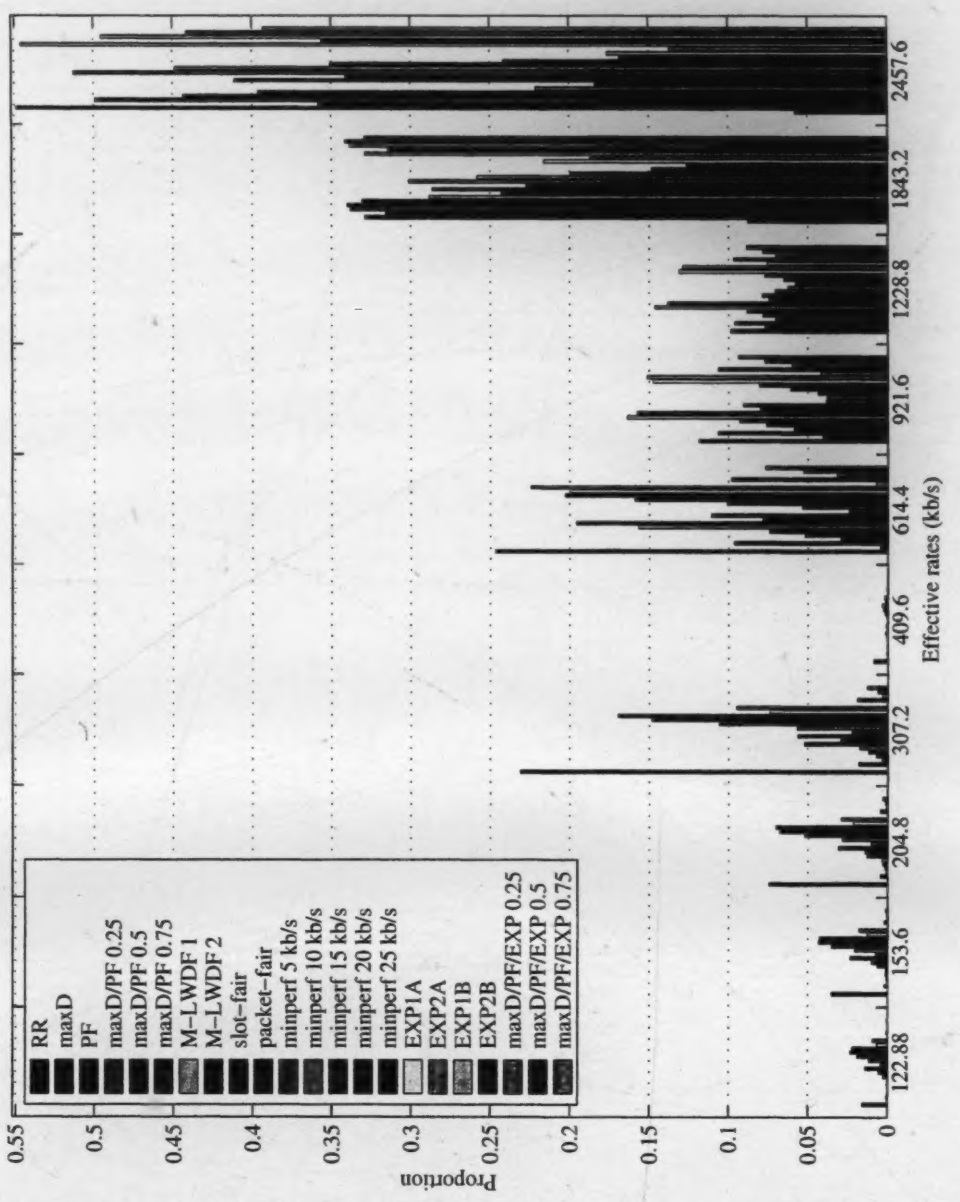
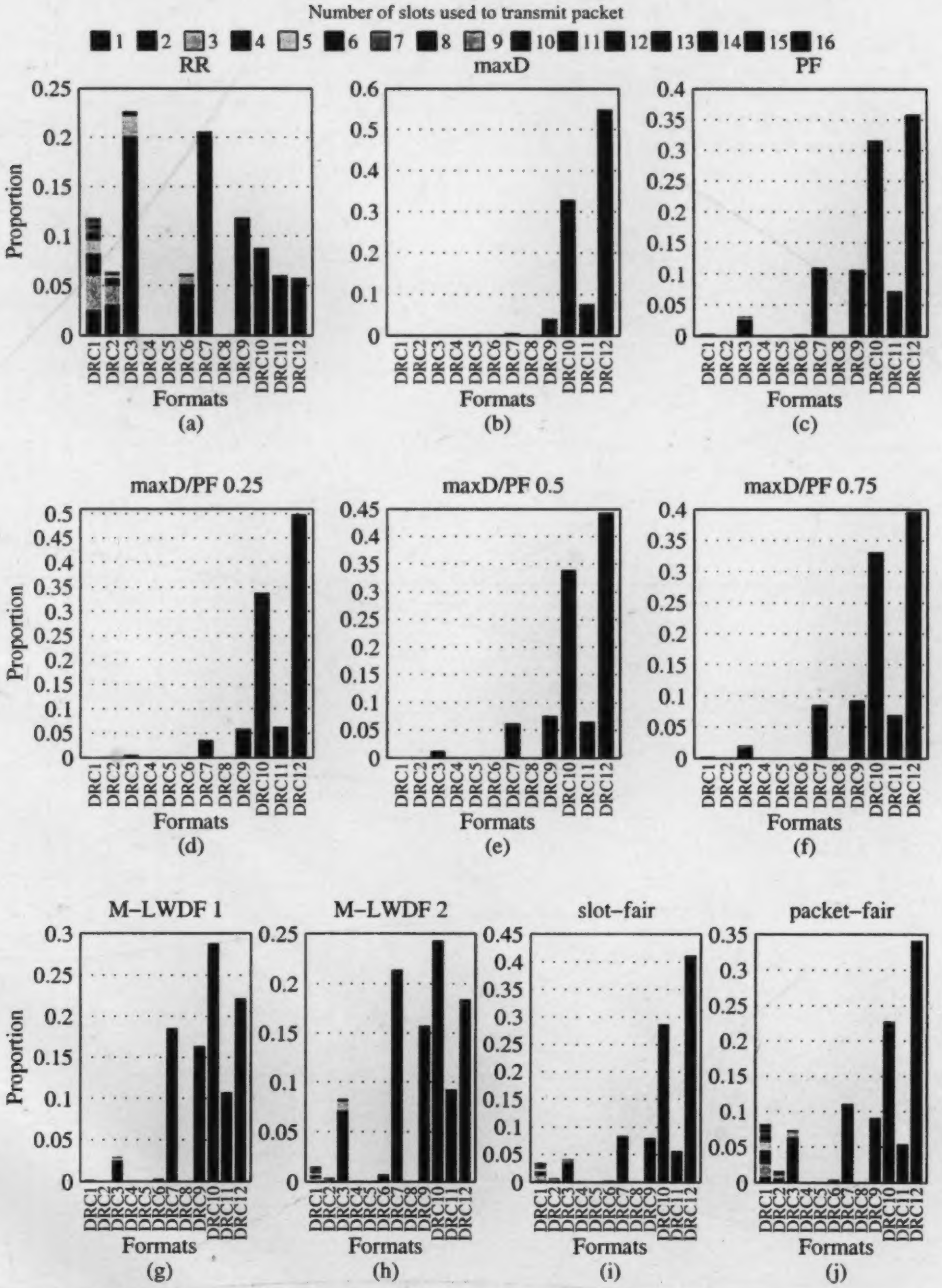
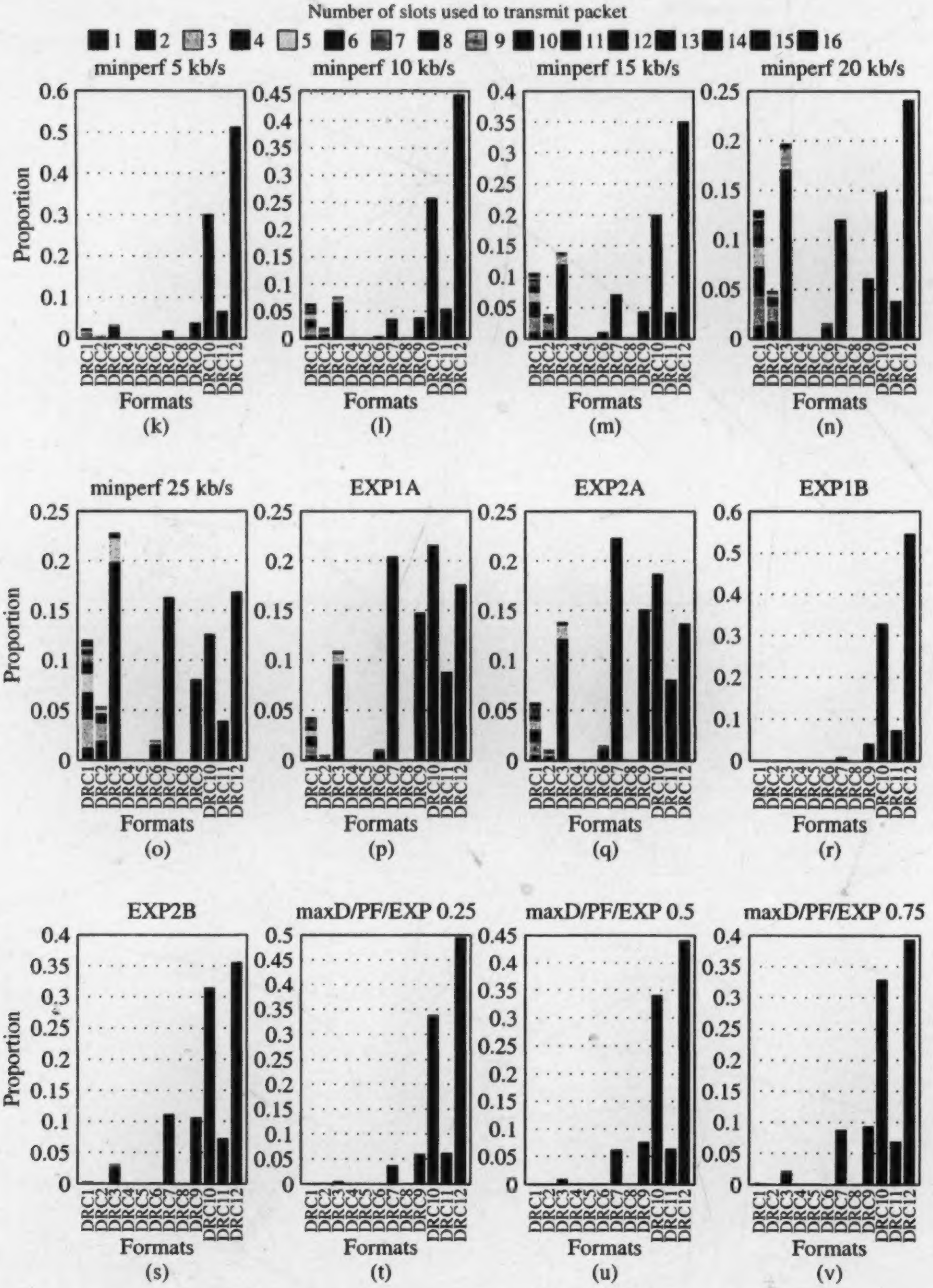


Figure G.3 (Part 2 of 2): Distribution of effective bit rates for transmitted packets in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure G.4 (Part 1 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins



(k) (l) (m) (n) (o) (p) (q) (r) (s) (t) (u) (v)

Figure G.4 (Part 2 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins

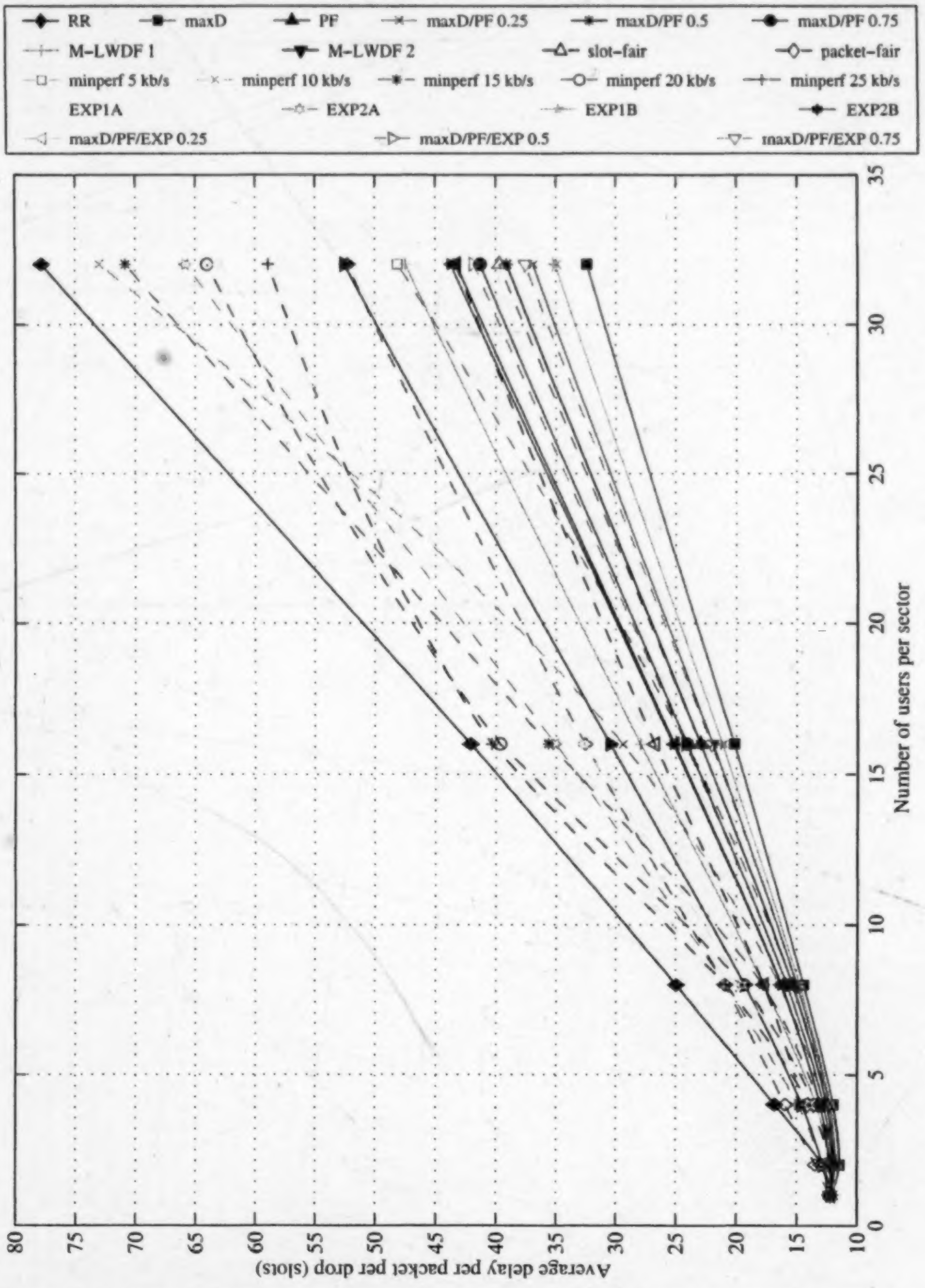


Figure G.5: Average delay per packet per drop versus number of users per sector in the ITU vehicular B channel using perfect one-slot prediction and margins

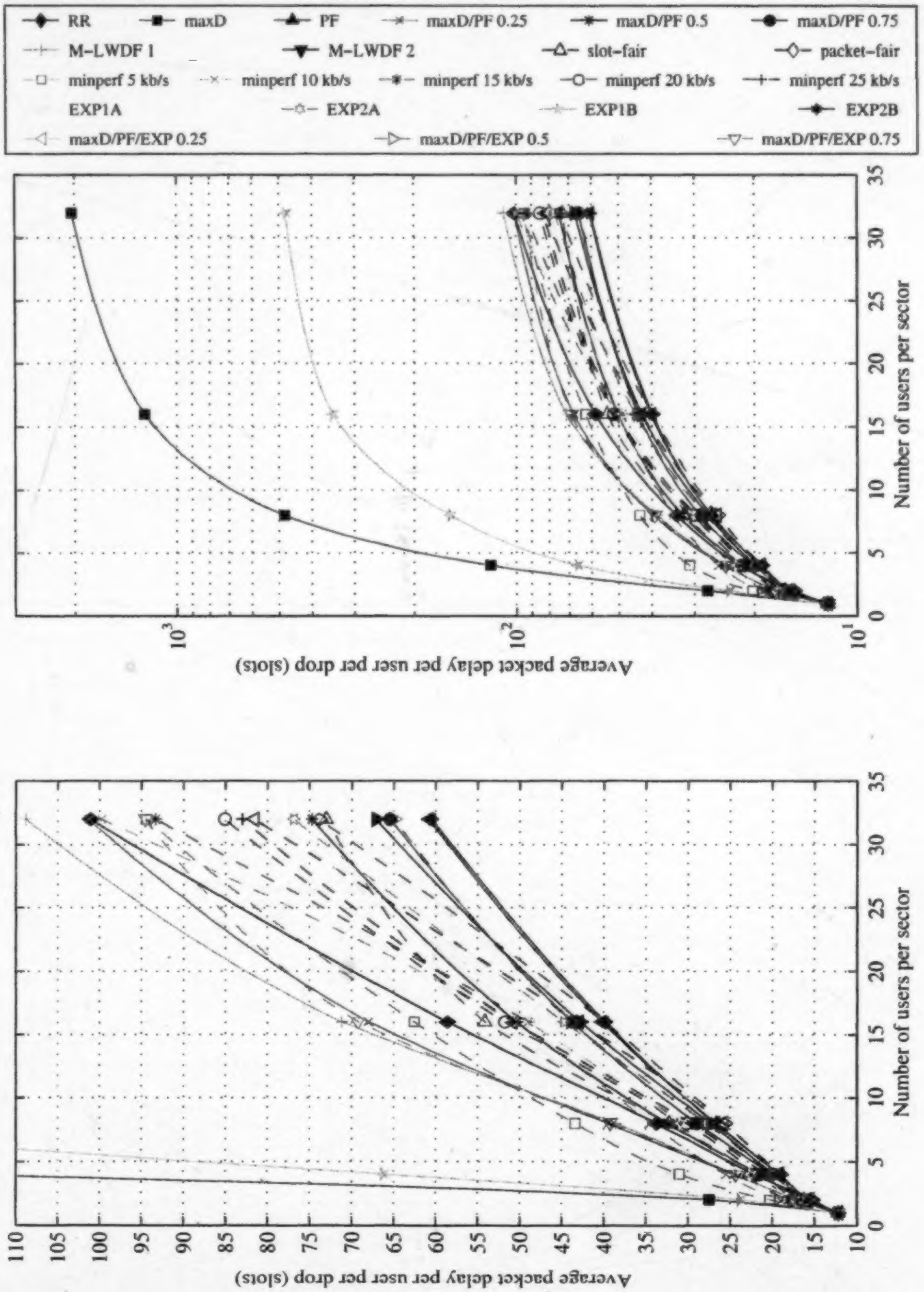


Figure G.6: Average packet delay per user per drop versus number of users per sector in the ITU vehicular B channel on a linear and a logarithmic scale using perfect one-slot prediction and margins

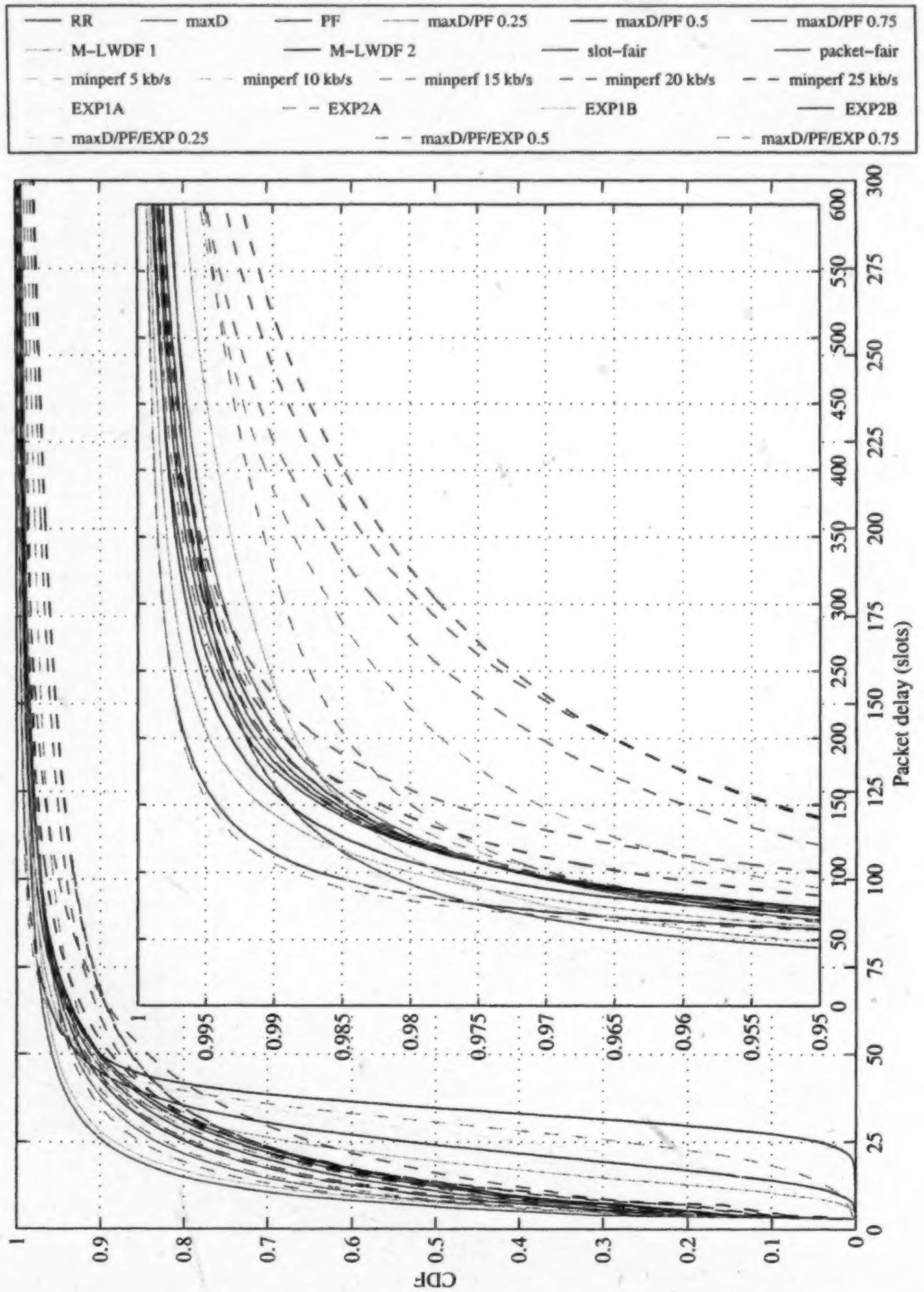


Figure G.7: Cumulative distribution functions of delays experienced by packets in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins

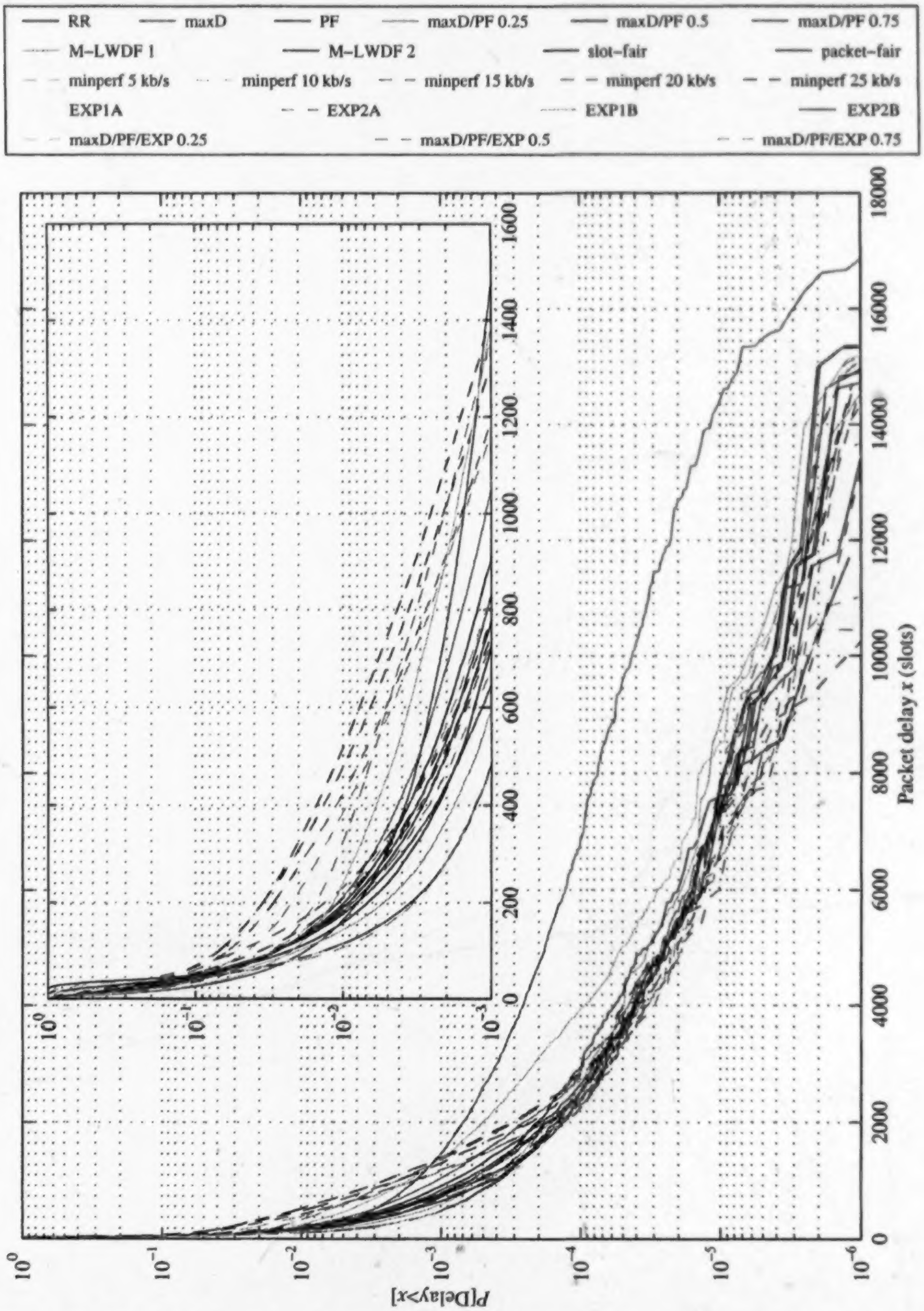


Figure G.8: Probability of delays experienced by packets being larger than a given value in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins

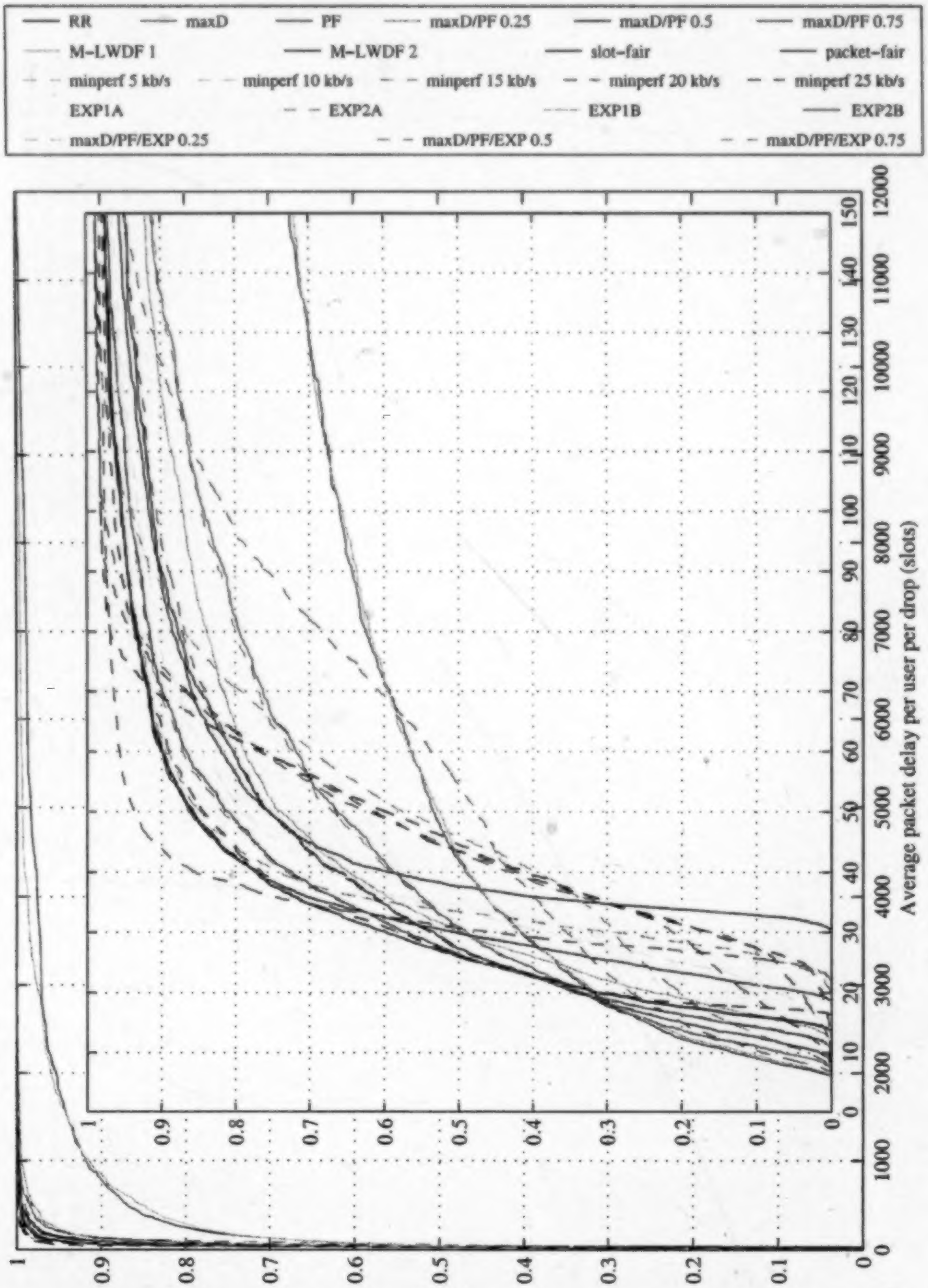


Figure G.9: Distributions of average packet delay per user for users who receive at least one packet in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins

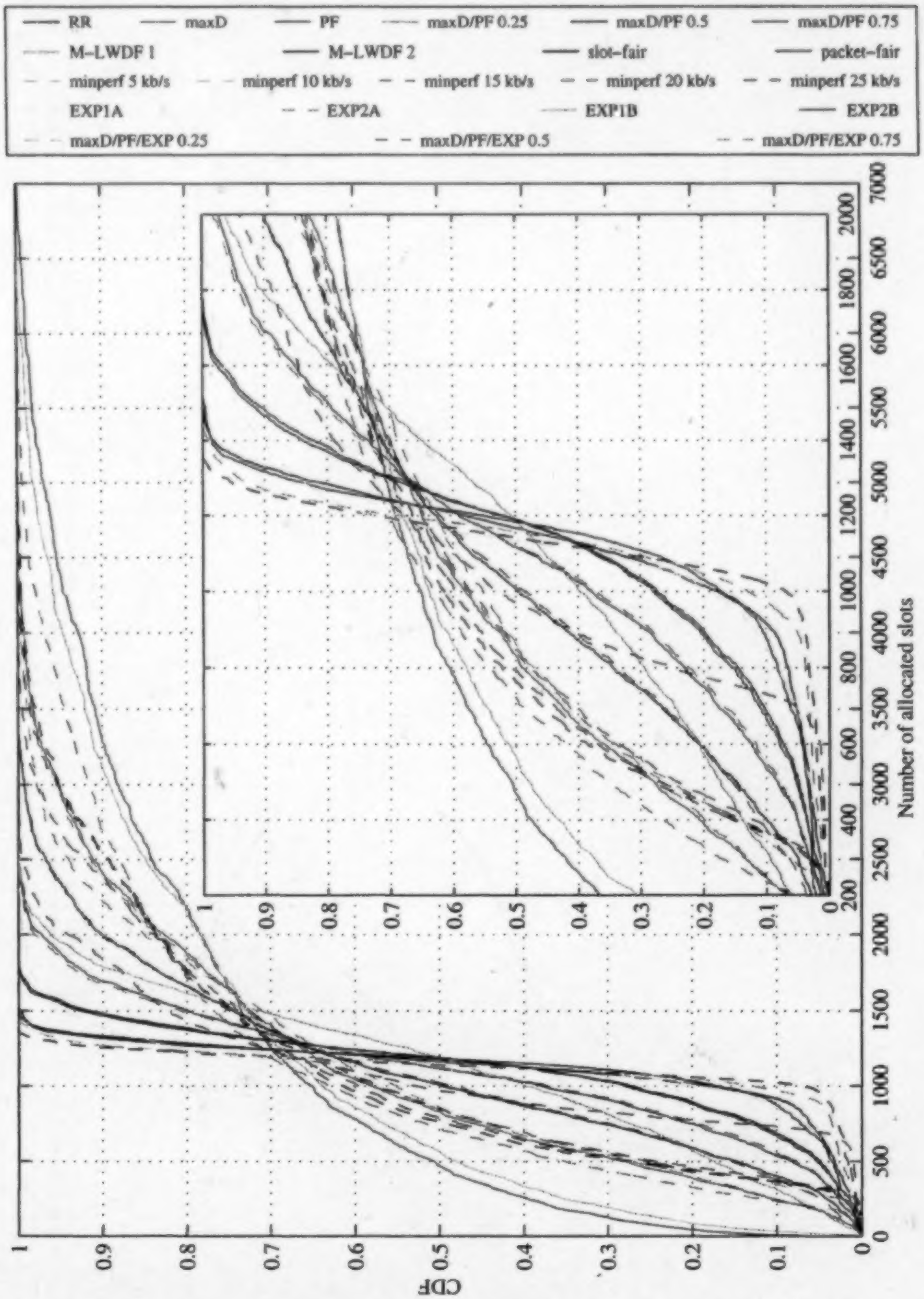


Figure G.10: Distributions of allocated slots per user per drop in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins

Appendix H: ITU Vehicular B Channel Results Comparing the "PF A" and "PF B" Average Rate Calculation Methods using Perfect One-slot Prediction and Margins

This appendix contains graphs comparing the simulation results using the "PF A" and "PF B" methods of calculating the average rate in the metrics of scheduling algorithms for the ITU vehicular B channel when perfect one-slot prediction plus margins was used to determine the transmission format each user could support.

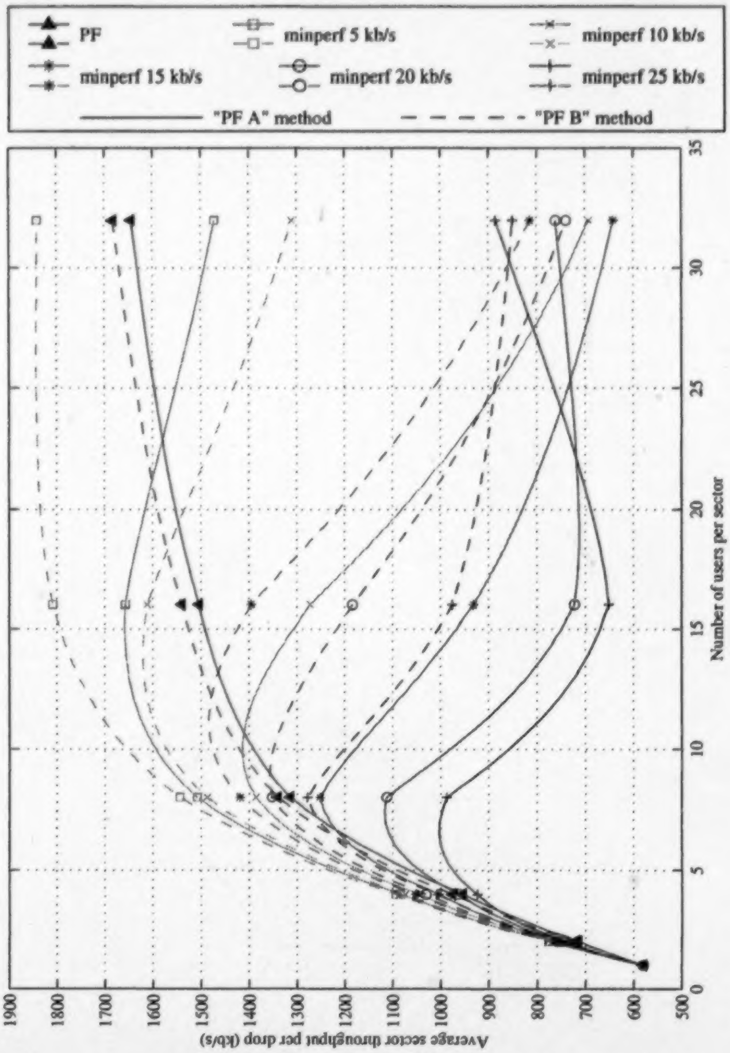


Figure H.1: Average sector throughput per drop vs. number of users per sector in the ITU vehicular B channel using perfect one-slot prediction and margins, comparing the "PF A" and "PF B" average rate calculation methods

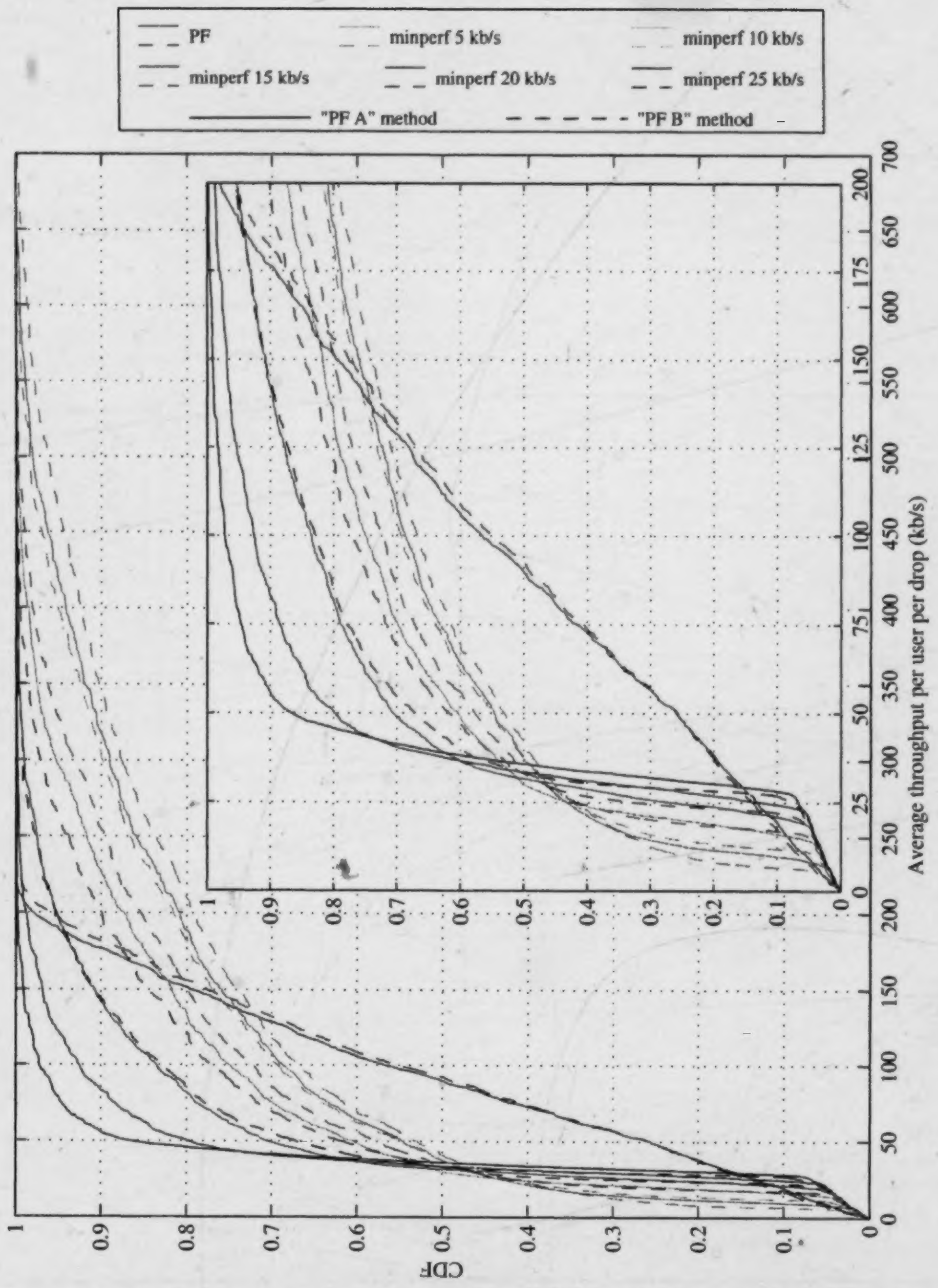


Figure H.2: Distributions of average throughput per user per drop in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins, comparing the "PF A" and "PF B" average rate calculation methods

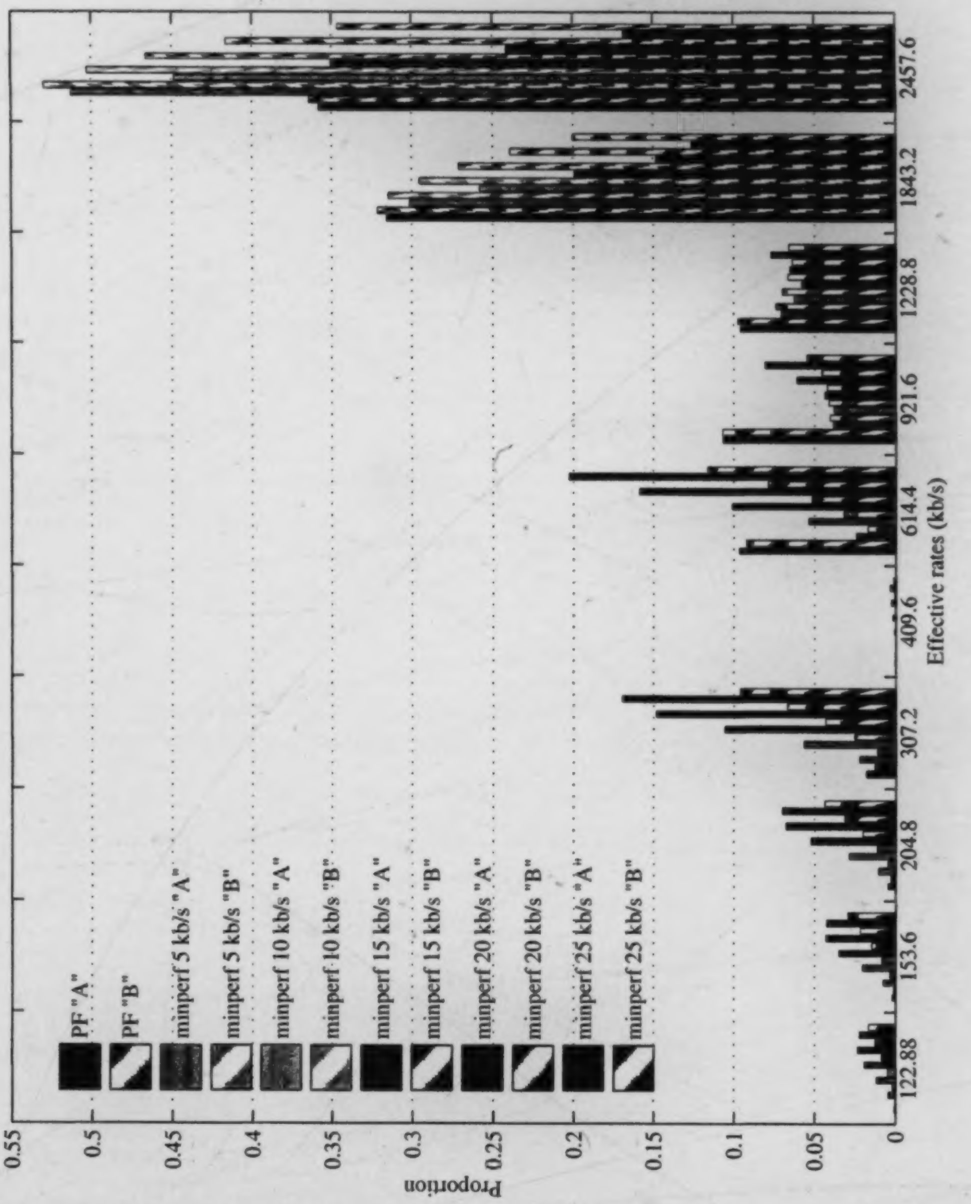


Figure H.3: Distribution of effective bit rates for transmitted packets in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins, comparing the "PF A" and "PF B" average rate calculation methods

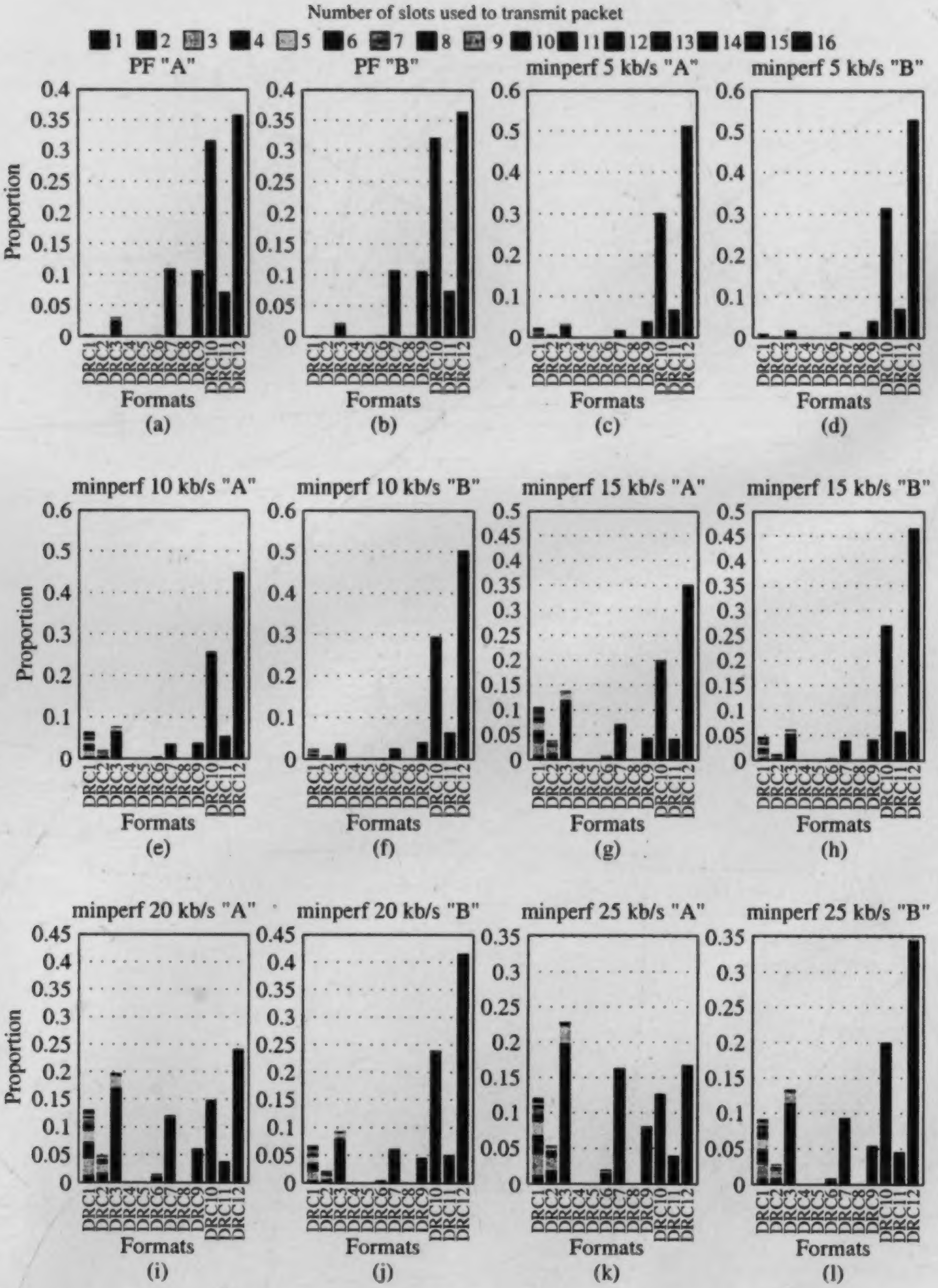


Figure H.4: Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins, comparing the "PF A" and "PF B" average rate calculation methods

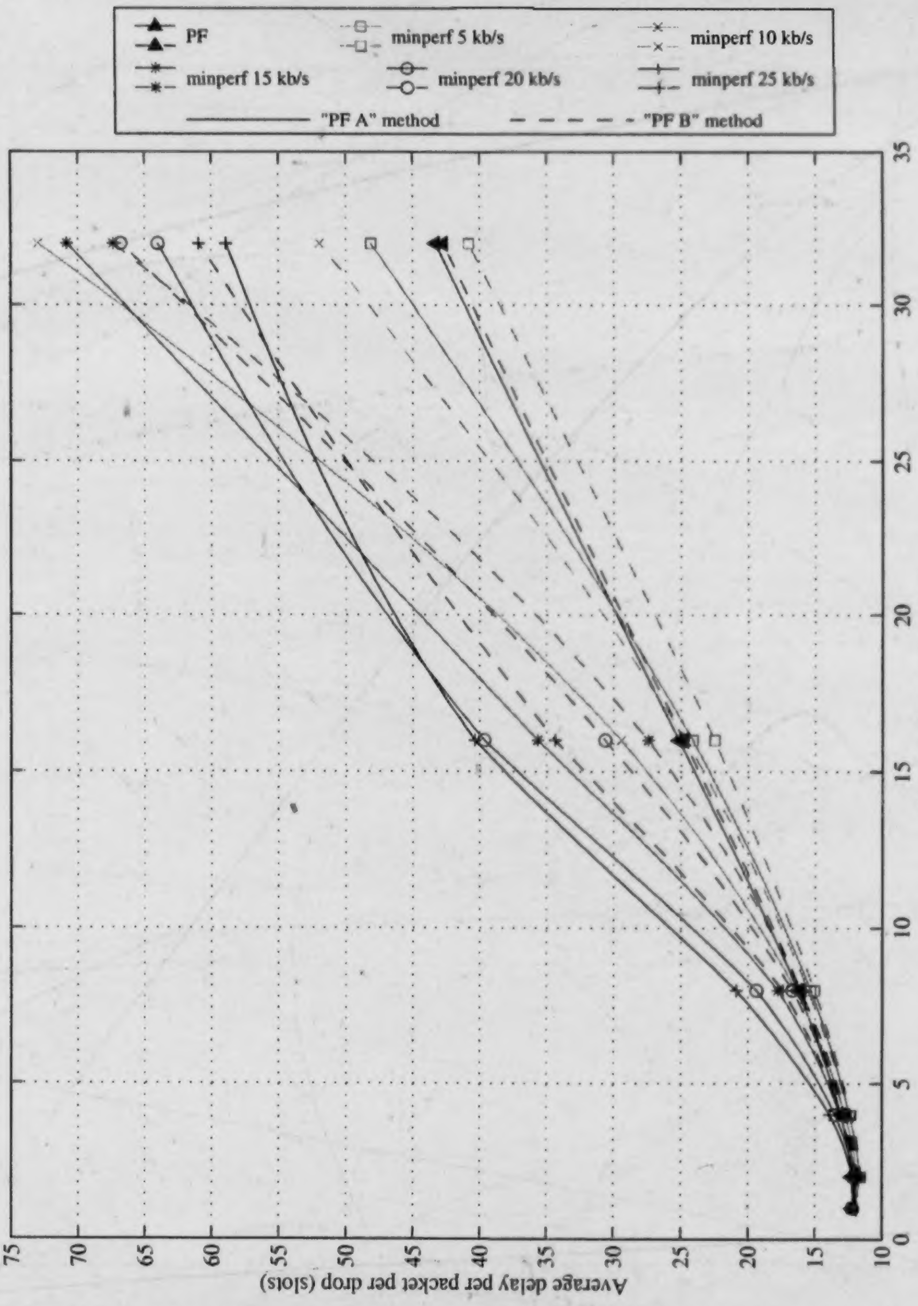


Figure H.5: Average delay per packet per drop versus number of users per sector in the ITU vehicular B channel using perfect one-slot prediction and margins, comparing the "PF A" and "PF B" average rate calculation methods

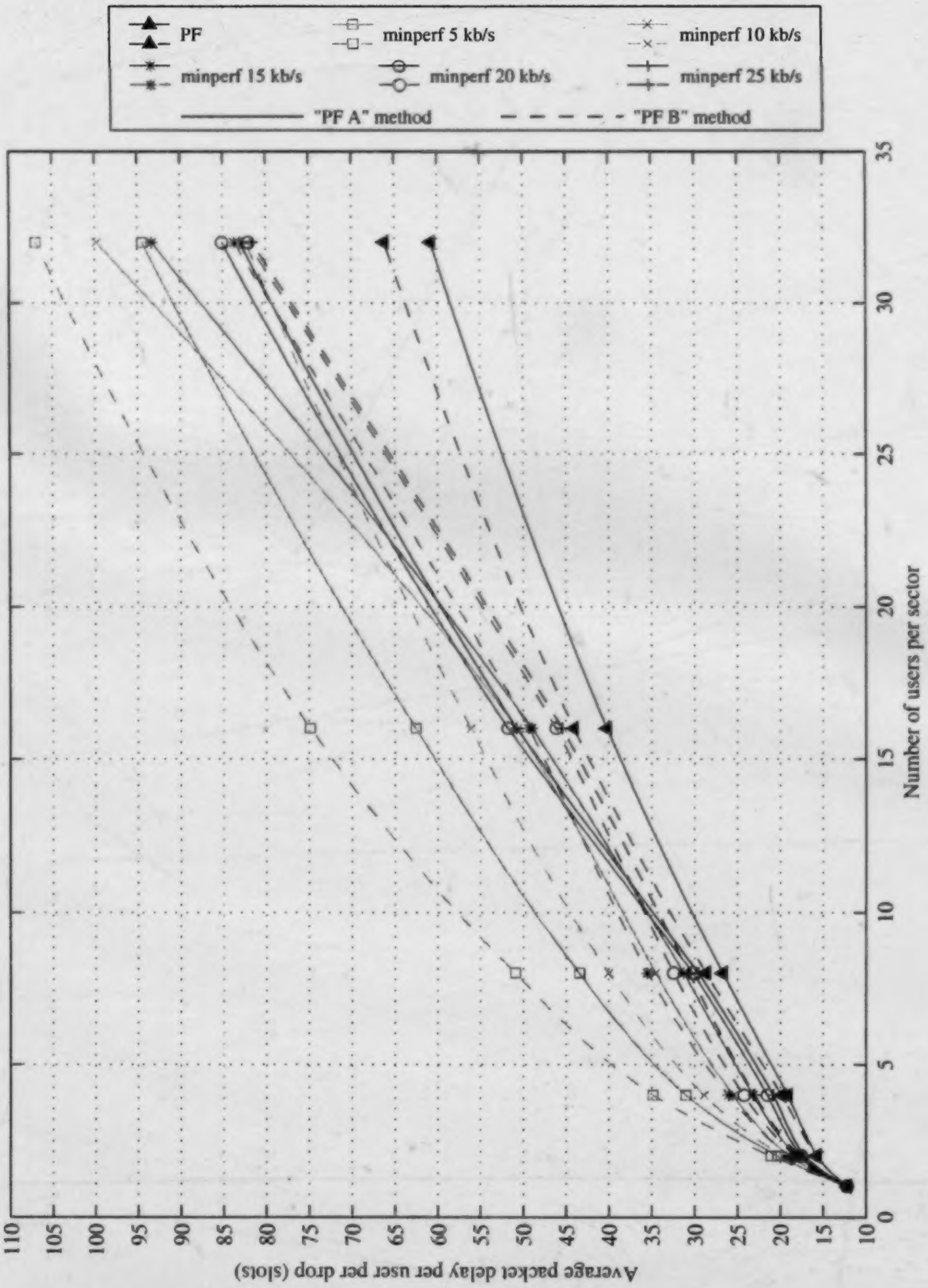


Figure H.6: Average packet delay per user per drop versus number of users per sector in the ITU vehicular B channel using perfect one-slot prediction and margins, comparing the "PF A" and "PF B" average rate calculation methods

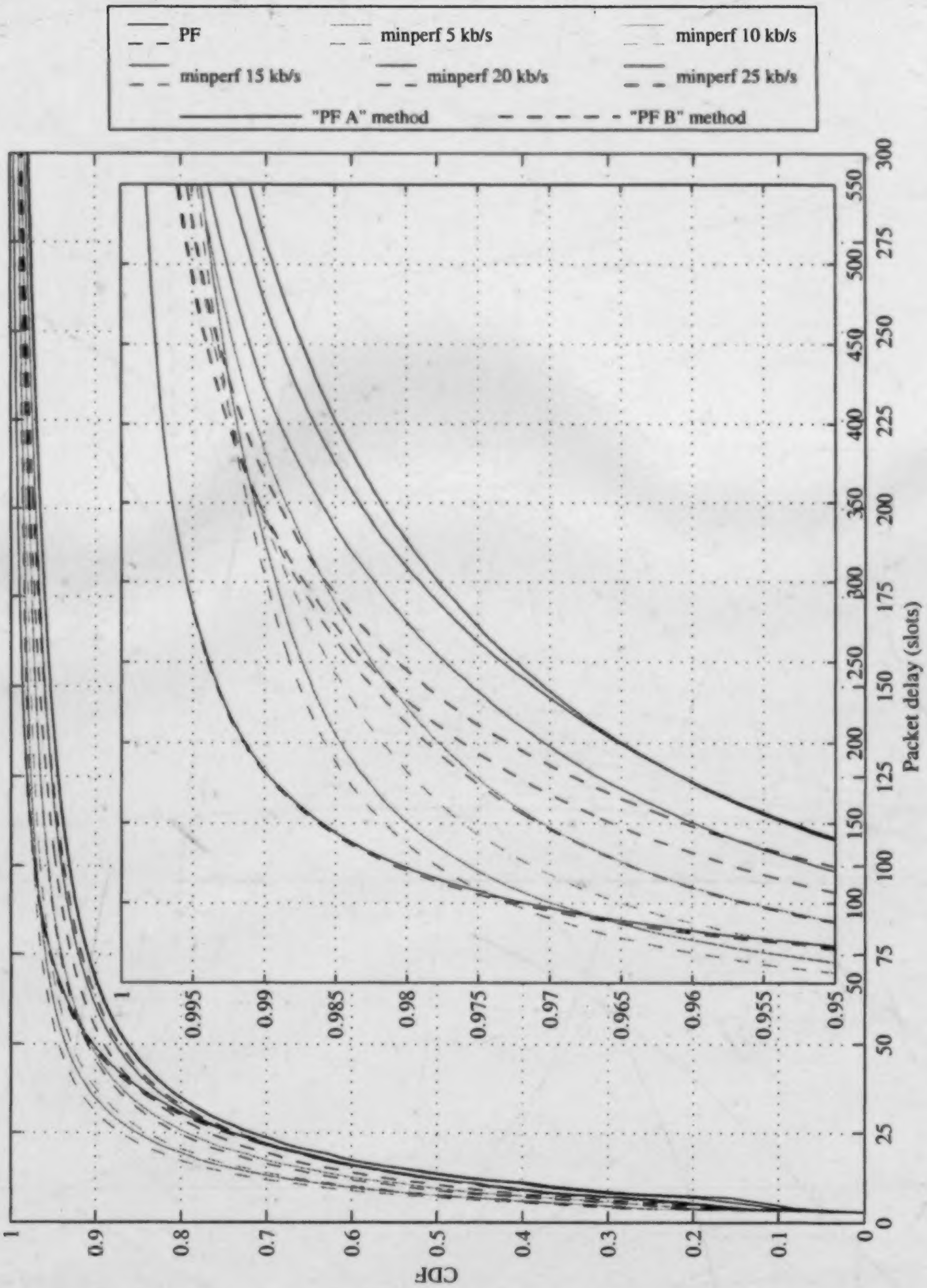


Figure H.7: Cumulative distribution functions of delays experienced by packets in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins, comparing the "PF A" and "PF B" average rate calculation methods

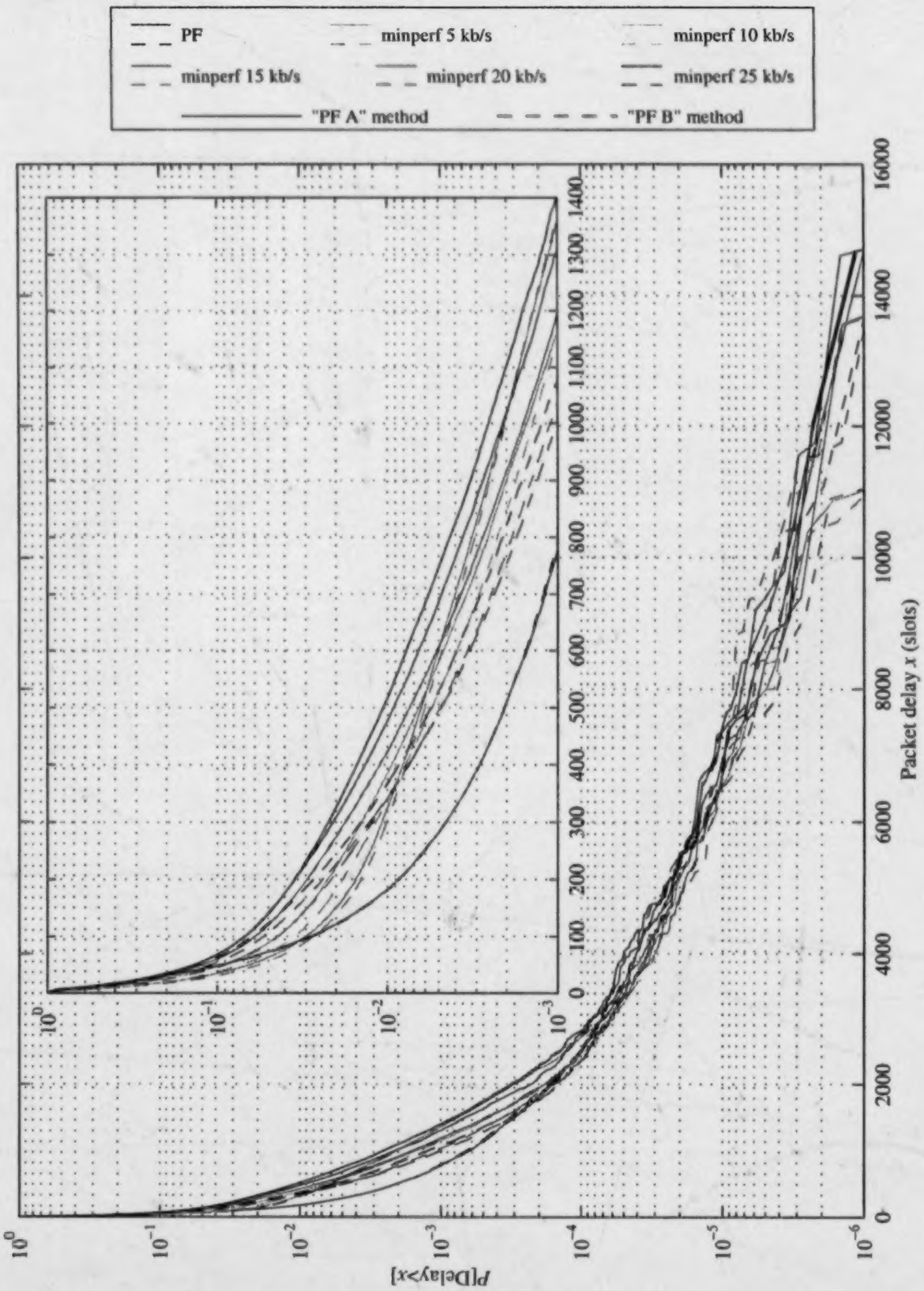


Figure H.8: Probability of delays experienced by packets being larger than a given value in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins and comparing the "PF A" and "PF B" average rate calculation methods

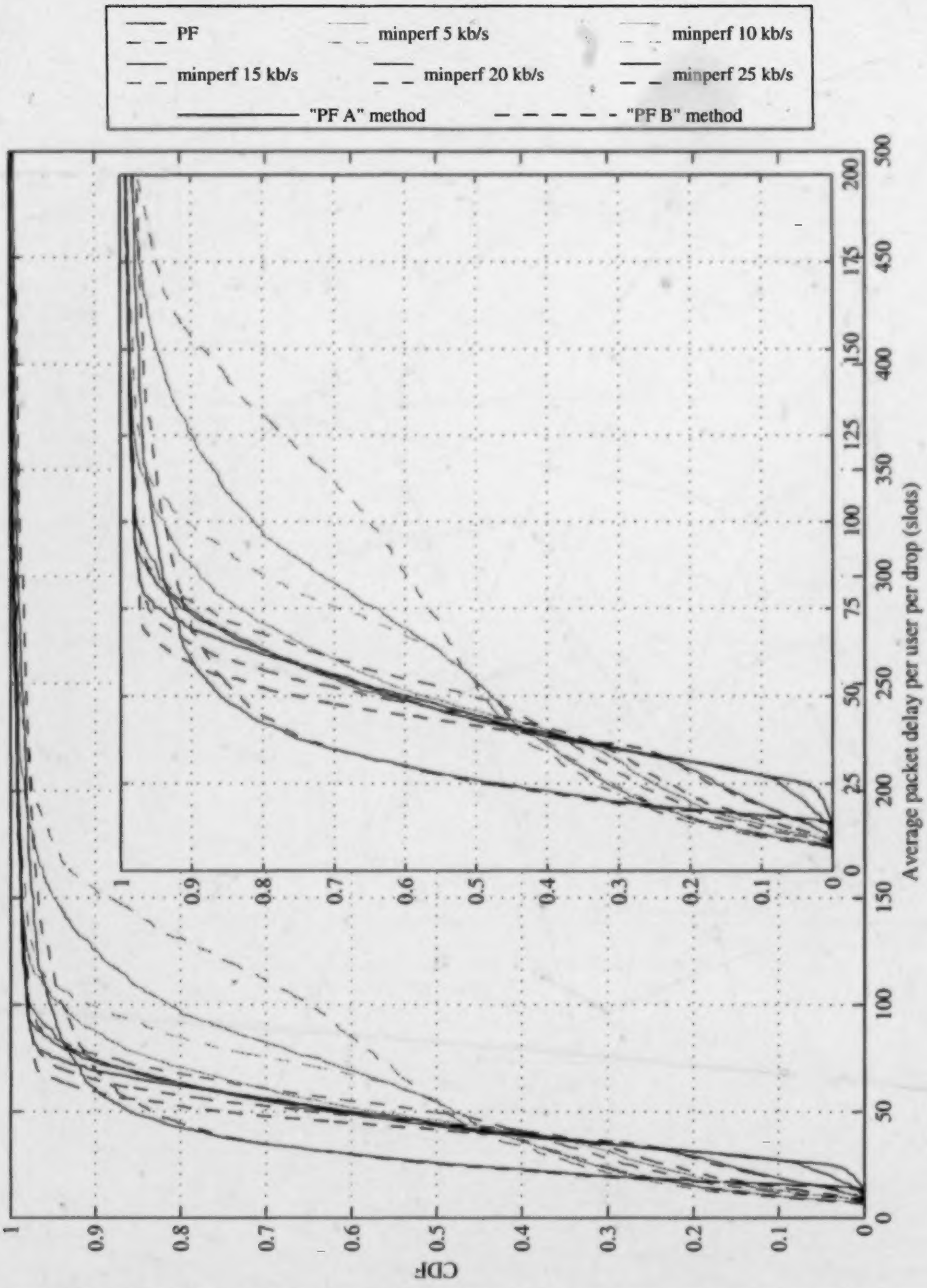


Figure H.9: Distributions of average packet delay per user for users who receive at least one packet in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins, comparing the "PF A" and "PF B" average rate calculation methods

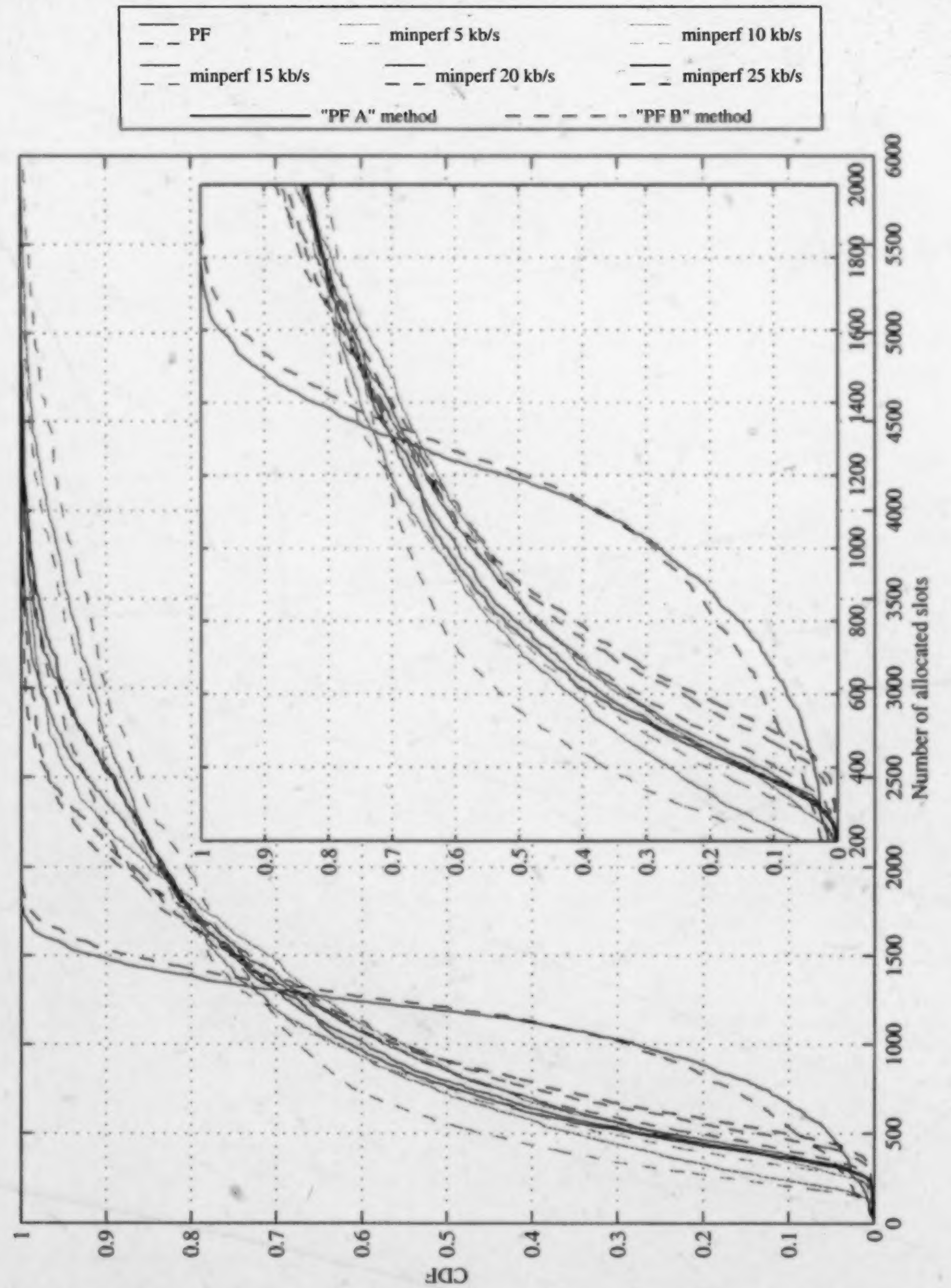


Figure H.10: Distributions of allocated slots per user per drop in the ITU vehicular B channel with 16 users per sector using perfect one-slot prediction and margins, comparing the “PF A” and “PF B” average rate calculation methods

Appendix I: ITU Vehicular B Channel Results using Perfect Multislot Prediction

This appendix contains graphs showing the simulation results for the ITU vehicular B channel when perfect multislot prediction was used to determine the transmission format each user could support.

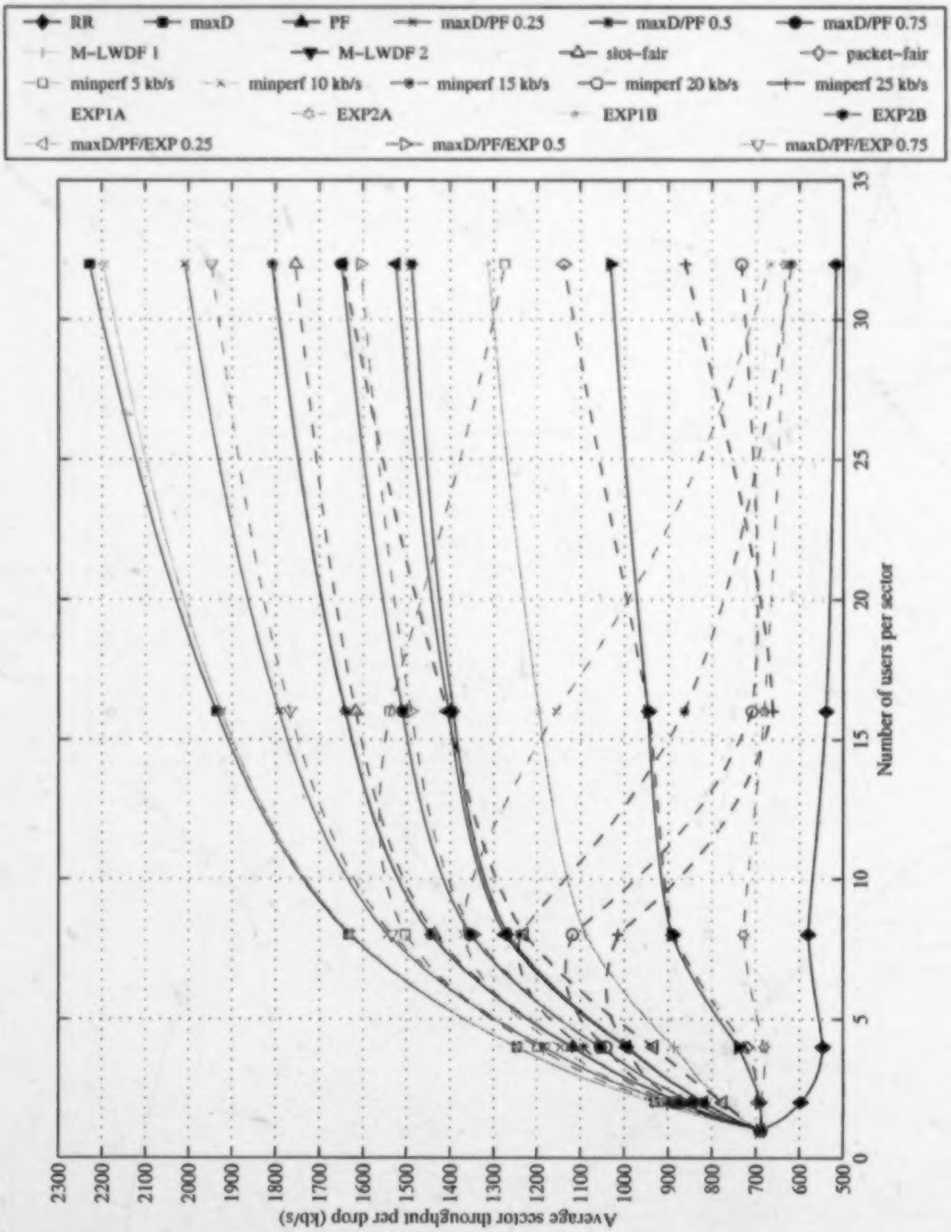


Figure I.1: Average sector throughput per drop vs. number of users per sector in the ITU vehicular B channel using perfect multislot prediction

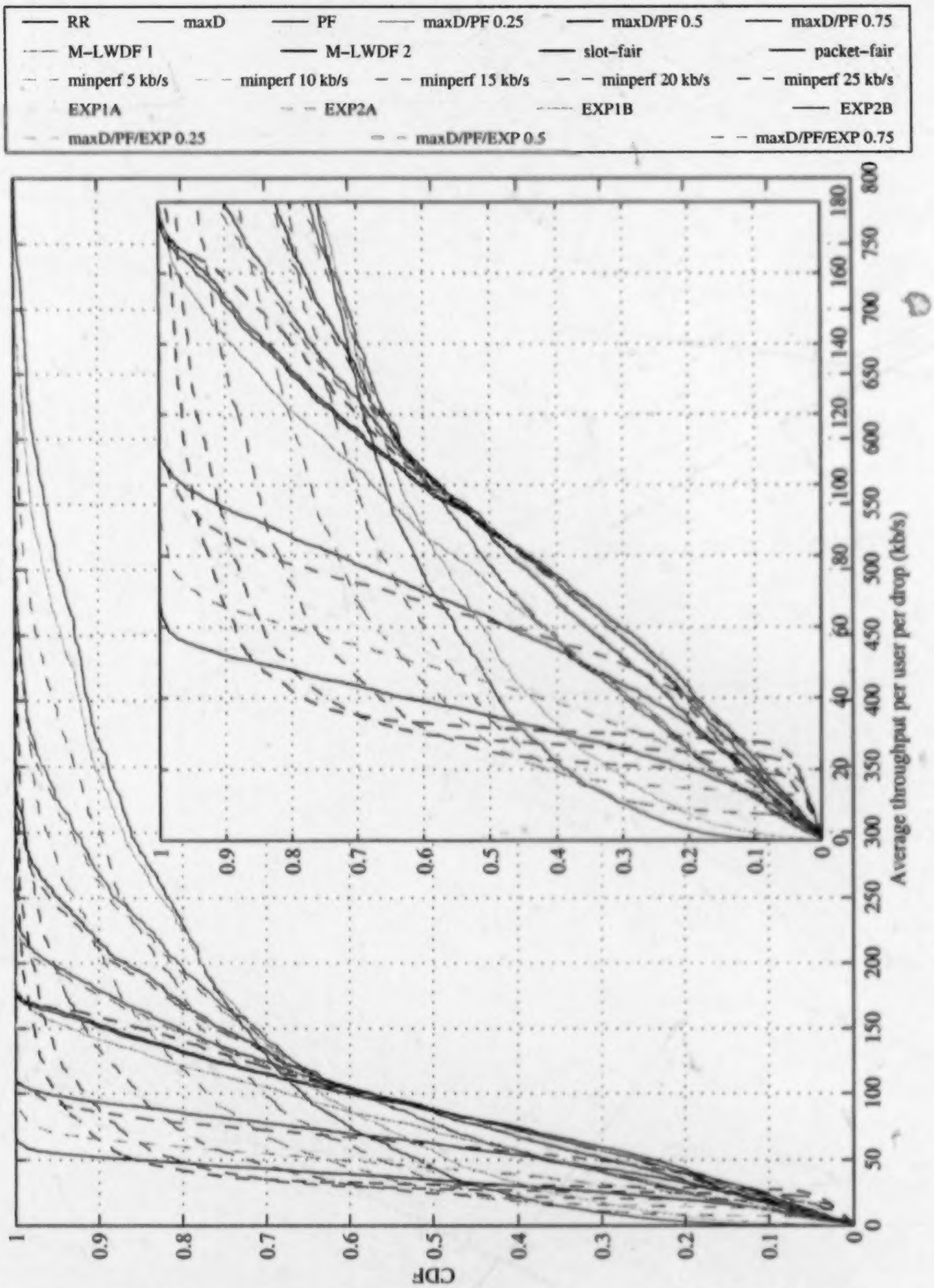


Figure I.2: Distributions of average throughput per user per drop in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction

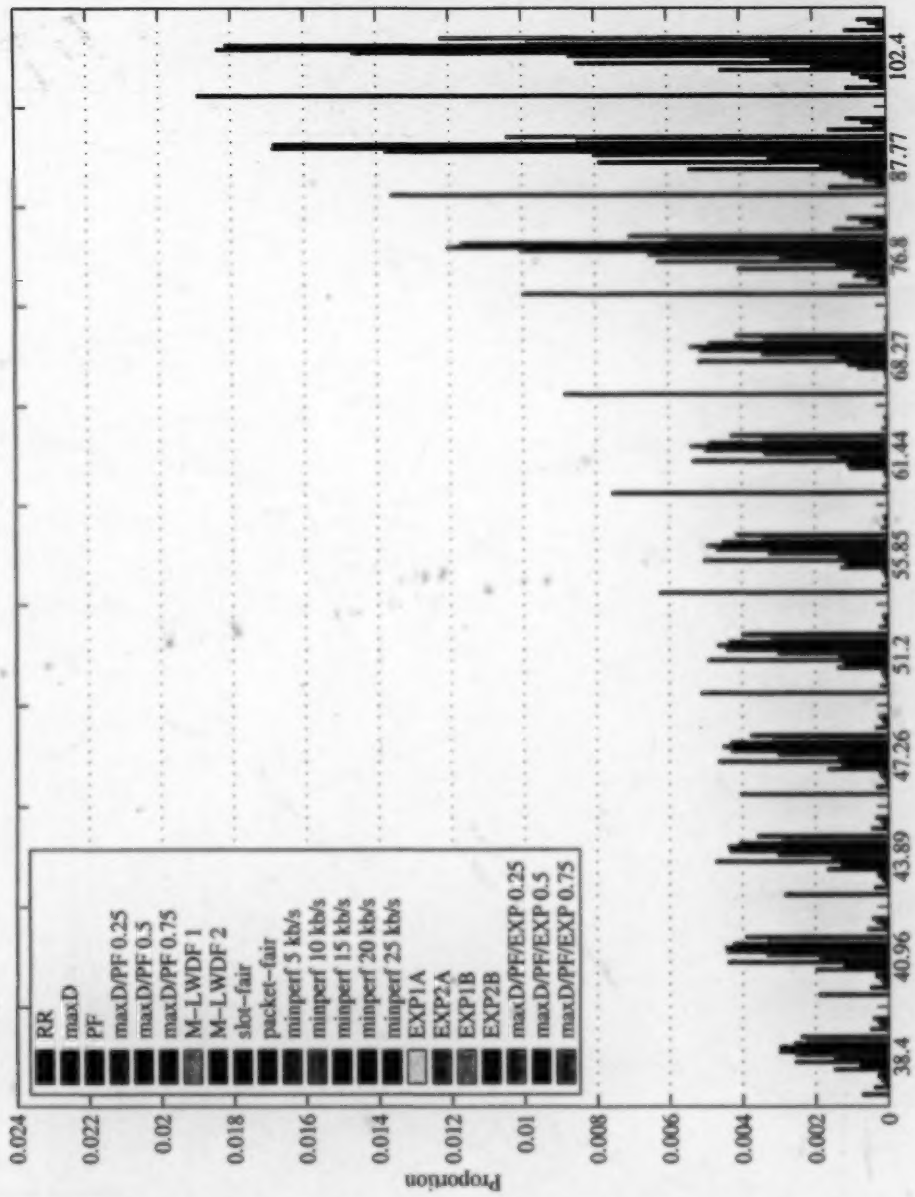


Figure I.3 (Part 1 of 2): Distribution of effective bit rates for transmitted packets in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction

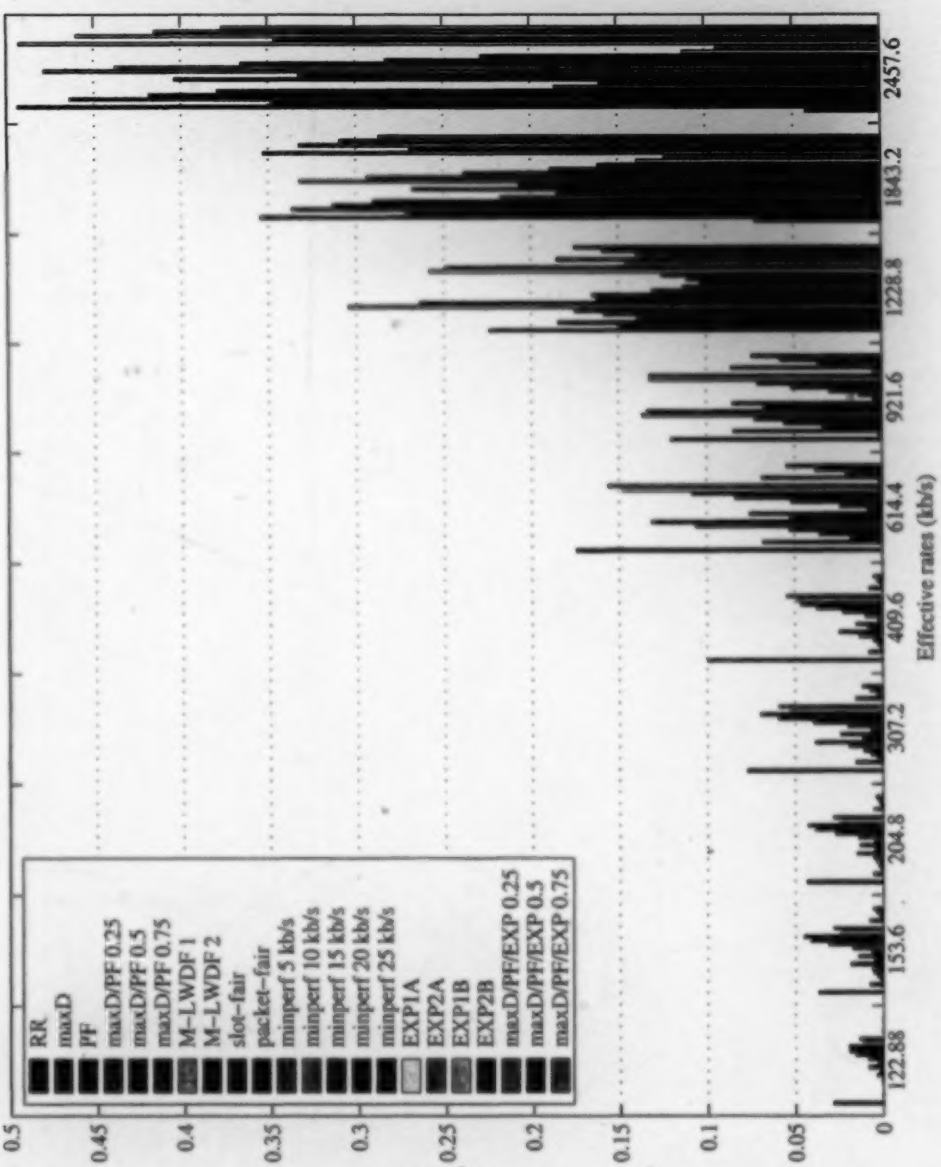
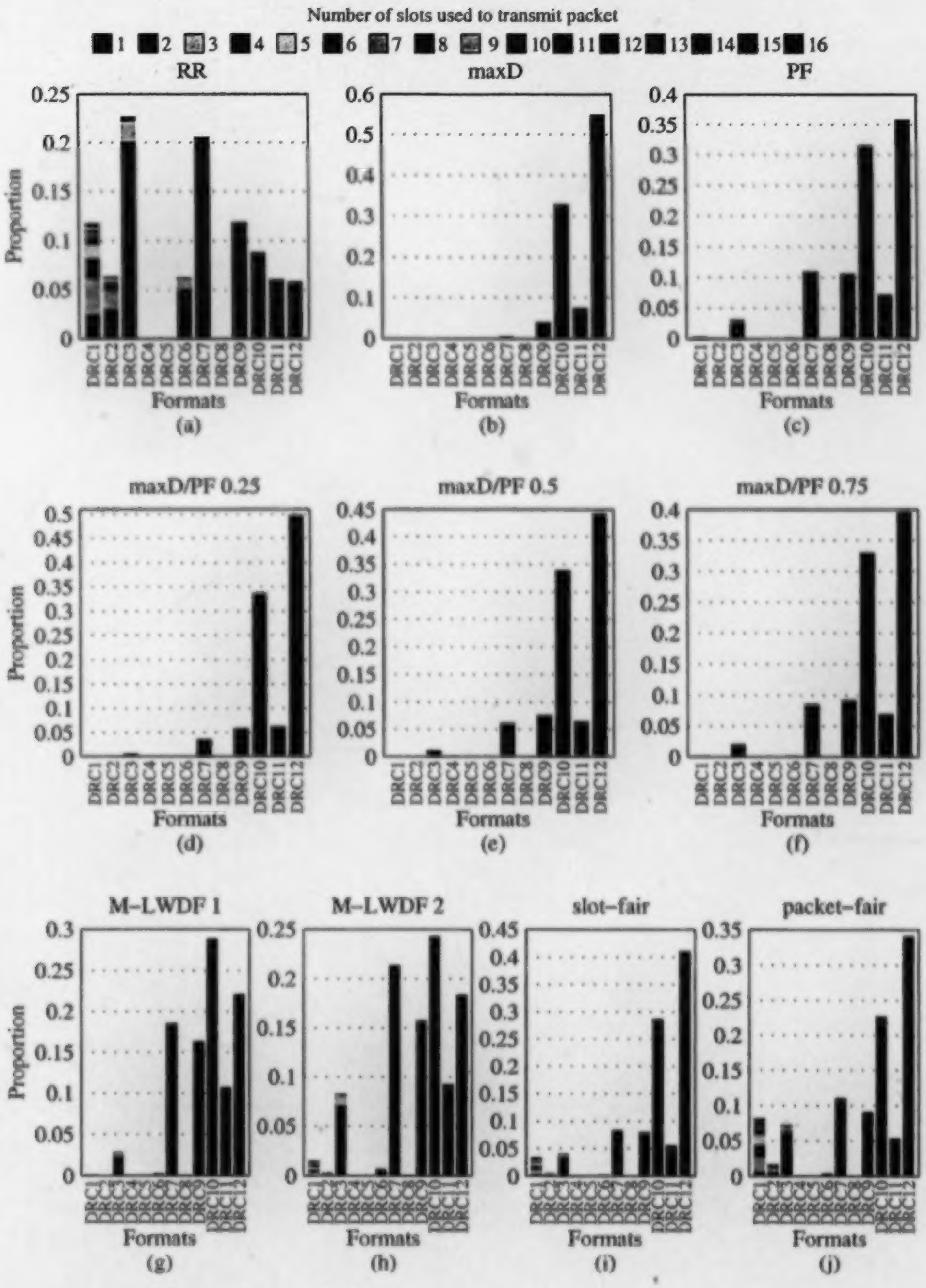


Figure I.3 (Part 2 of 2): Distribution of effective bit rates for transmitted packets in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction

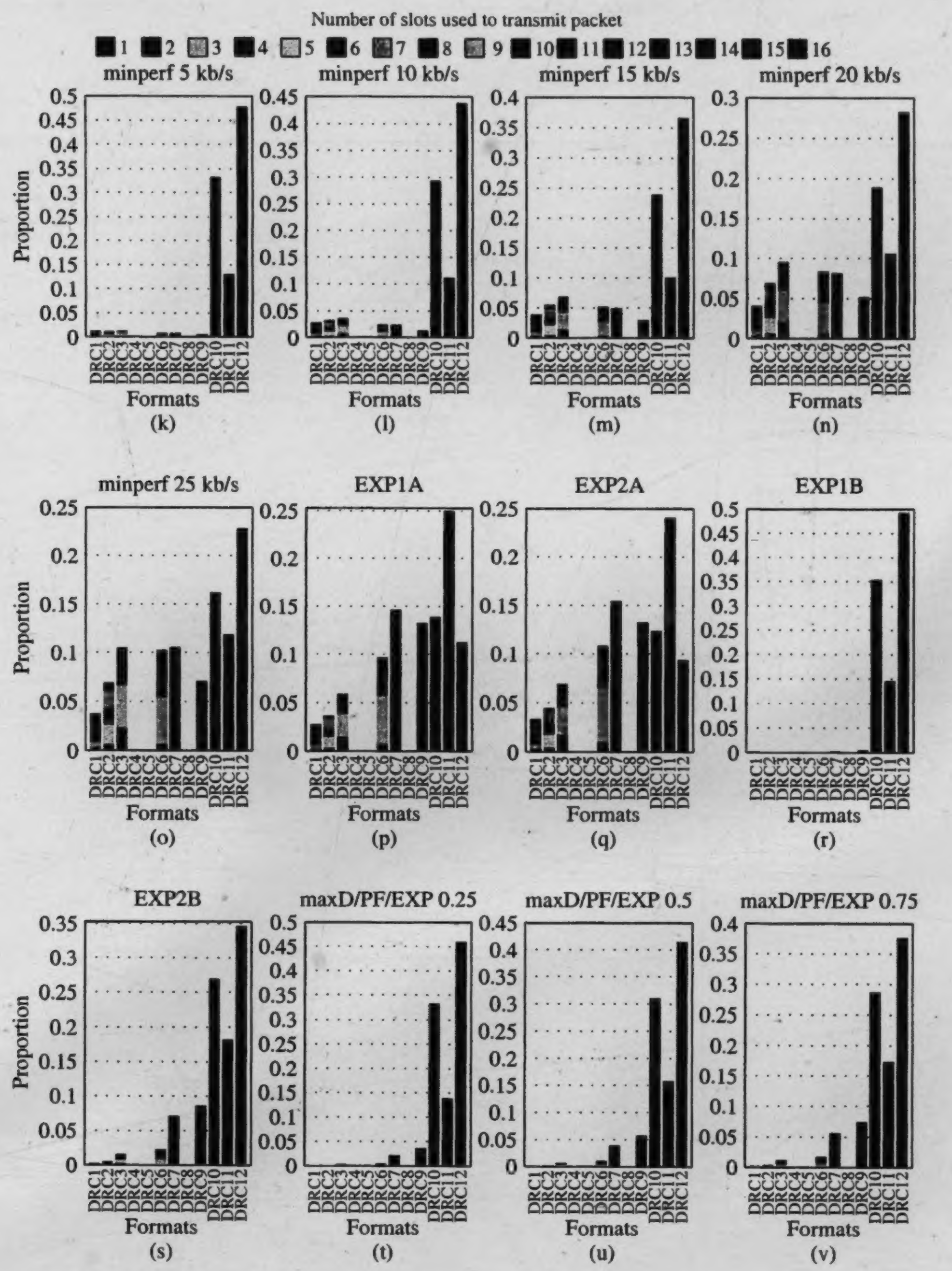


(a) (b) (c)

(d) (e) (f)

(g) (h) (i) (j)

Figure I.4 (Part 1 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction



(k) (l) (m) (n)

(o) (p) (q) (r)

(s) (t) (u) (v)

Figure I.4 (Part 2 of 2): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction

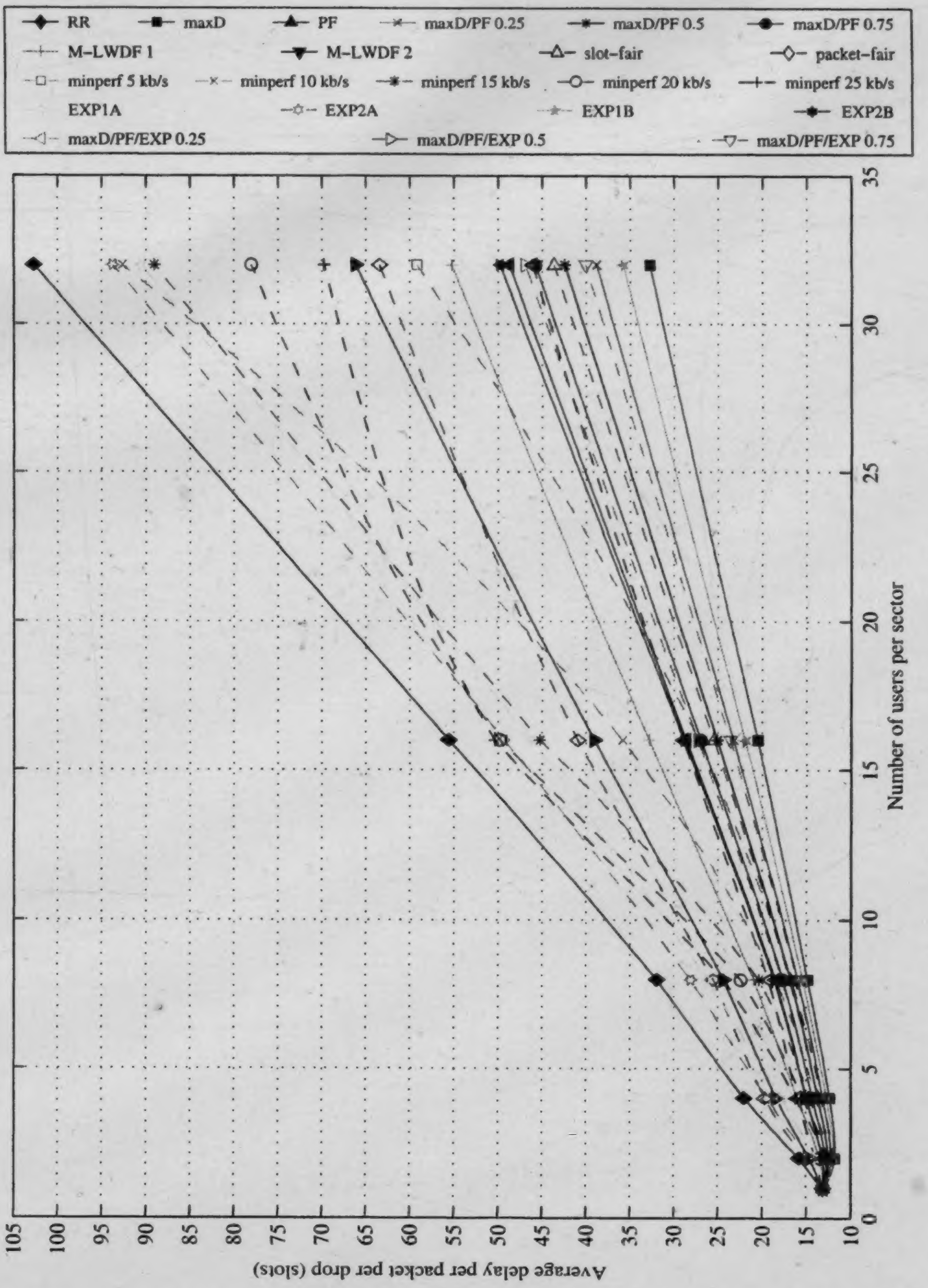


Figure I.5: Average delay per packet per drop versus number of users per sector in the ITU vehicular B channel using perfect multislot prediction

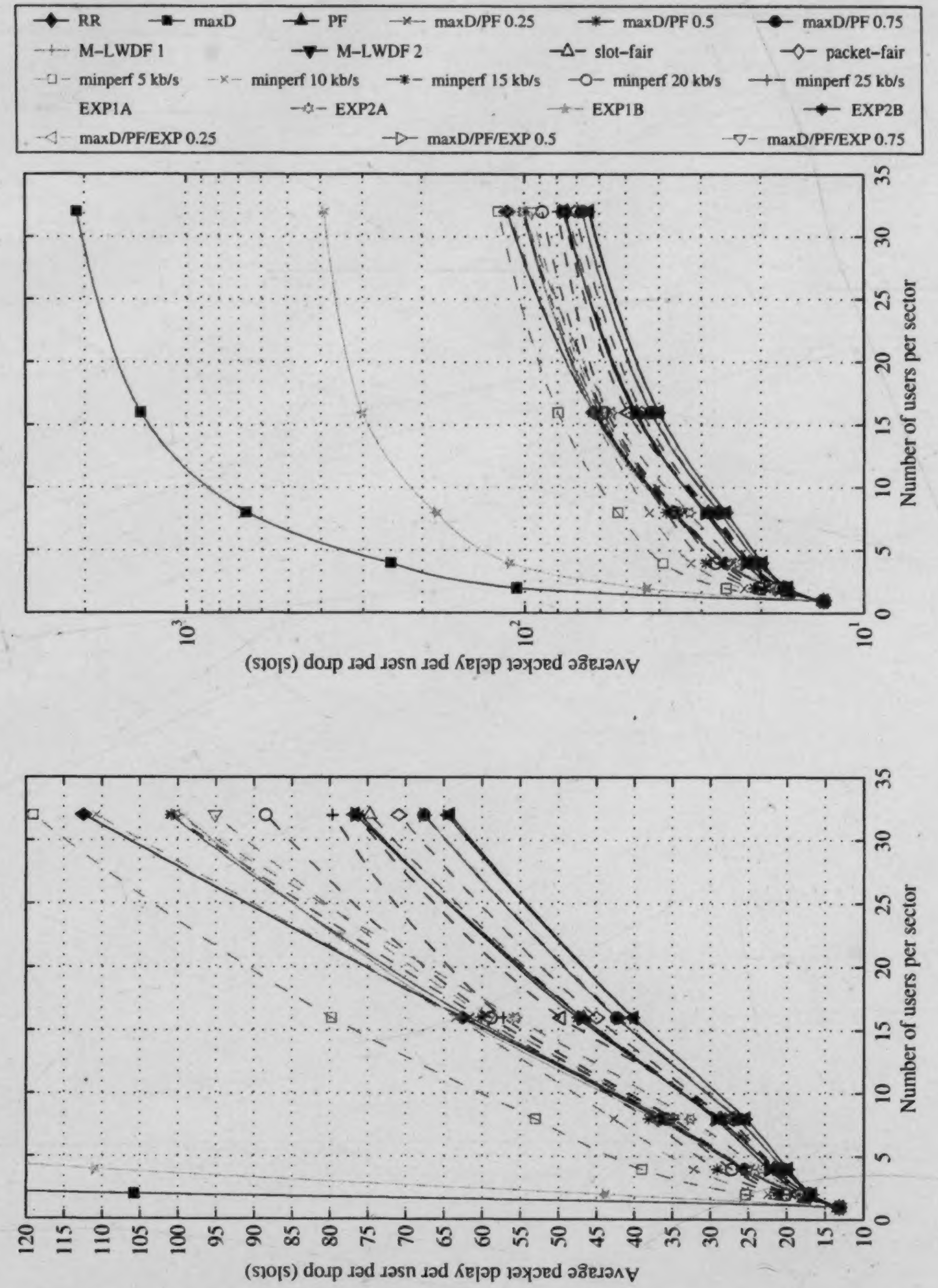


Figure I.6: Average packet delay per user per drop versus number of users per sector in the ITU vehicular B channel on a linear and a logarithmic scale using perfect multislot prediction

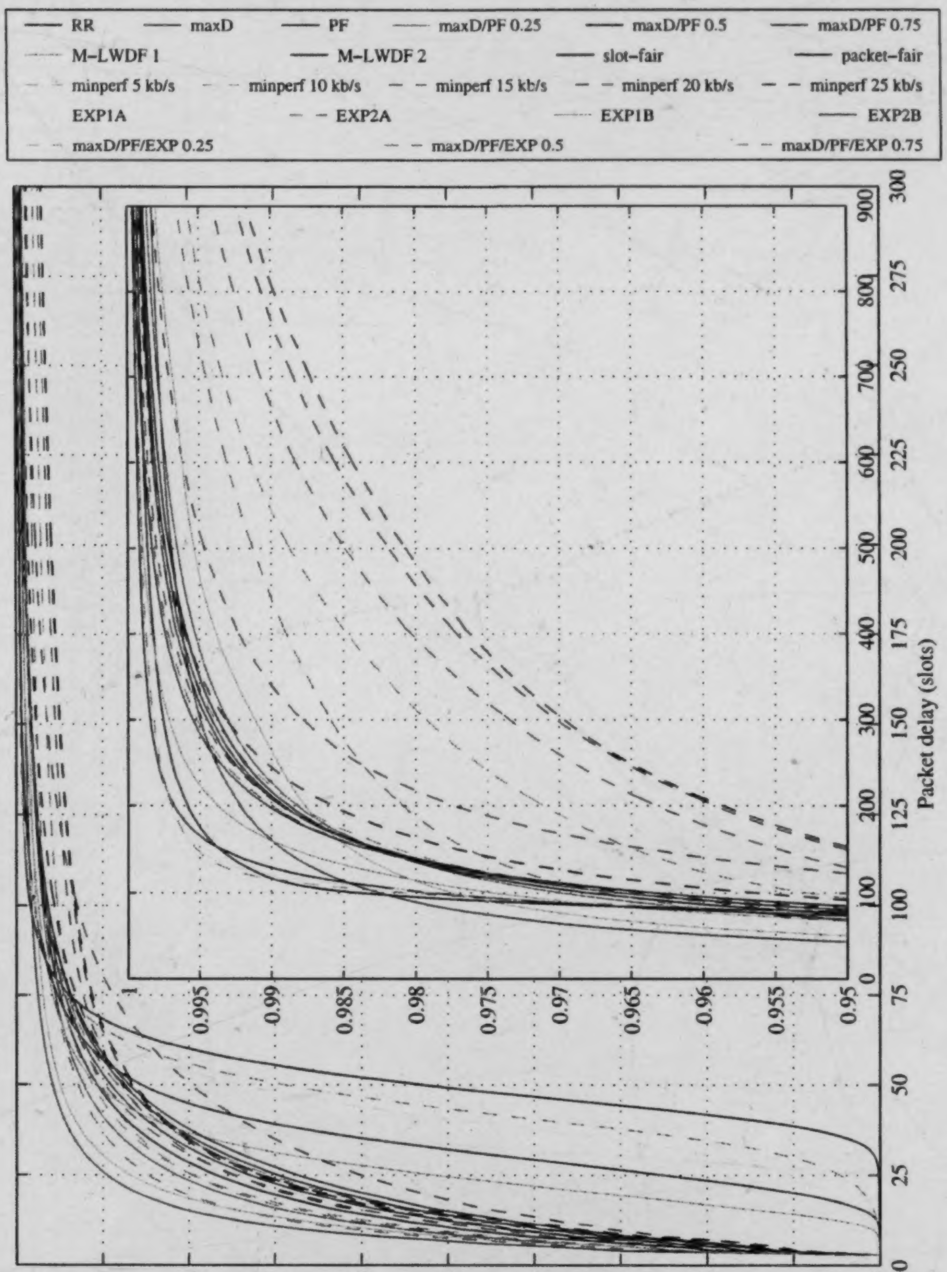


Figure I.7: Cumulative distribution functions of delays experienced by packets in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction

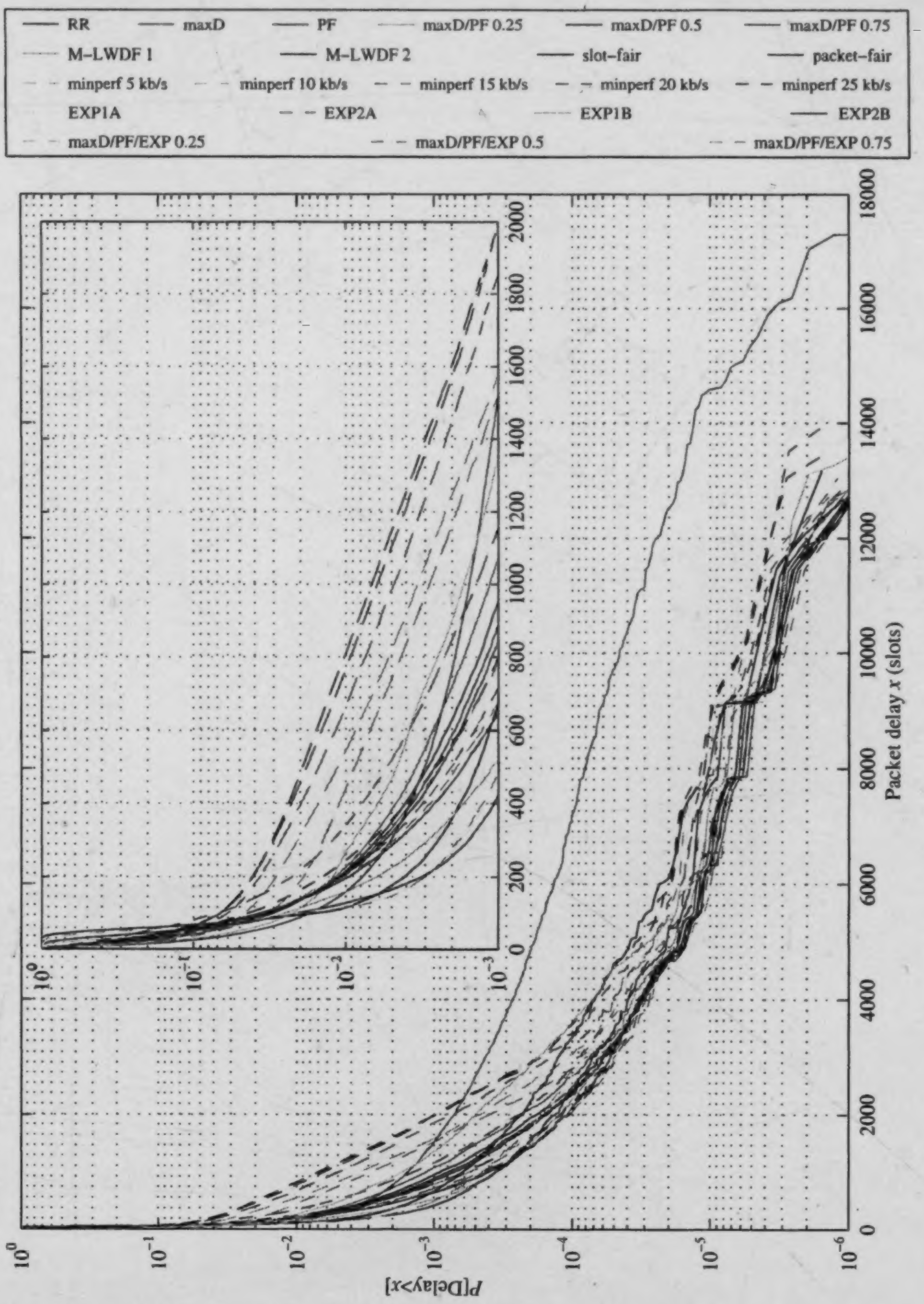


Figure I.8: Probability of delays experienced by packets being larger than a given value in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction

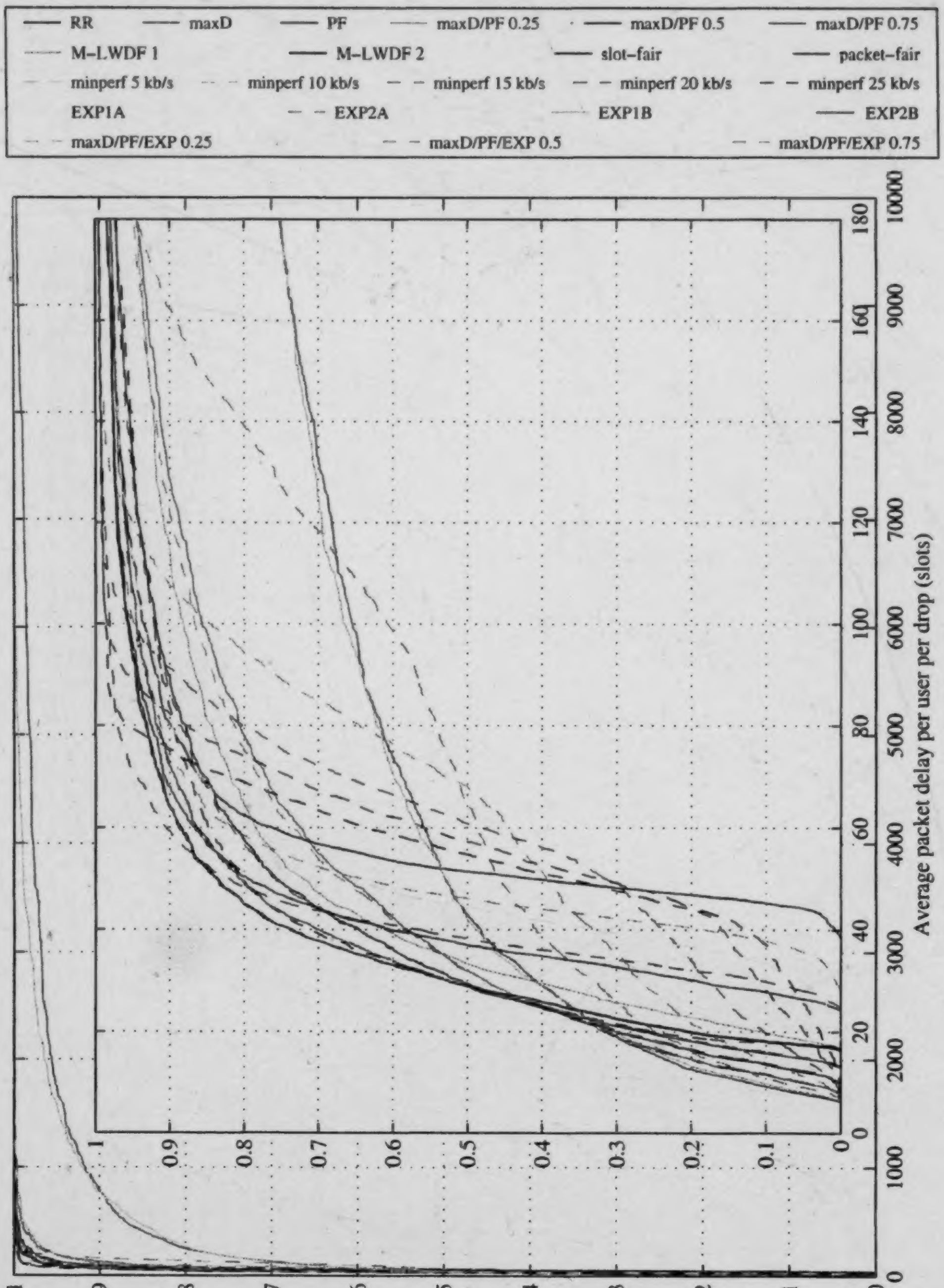


Figure I.9: Distributions of average packet delay per user for users who receive at least one packet in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction

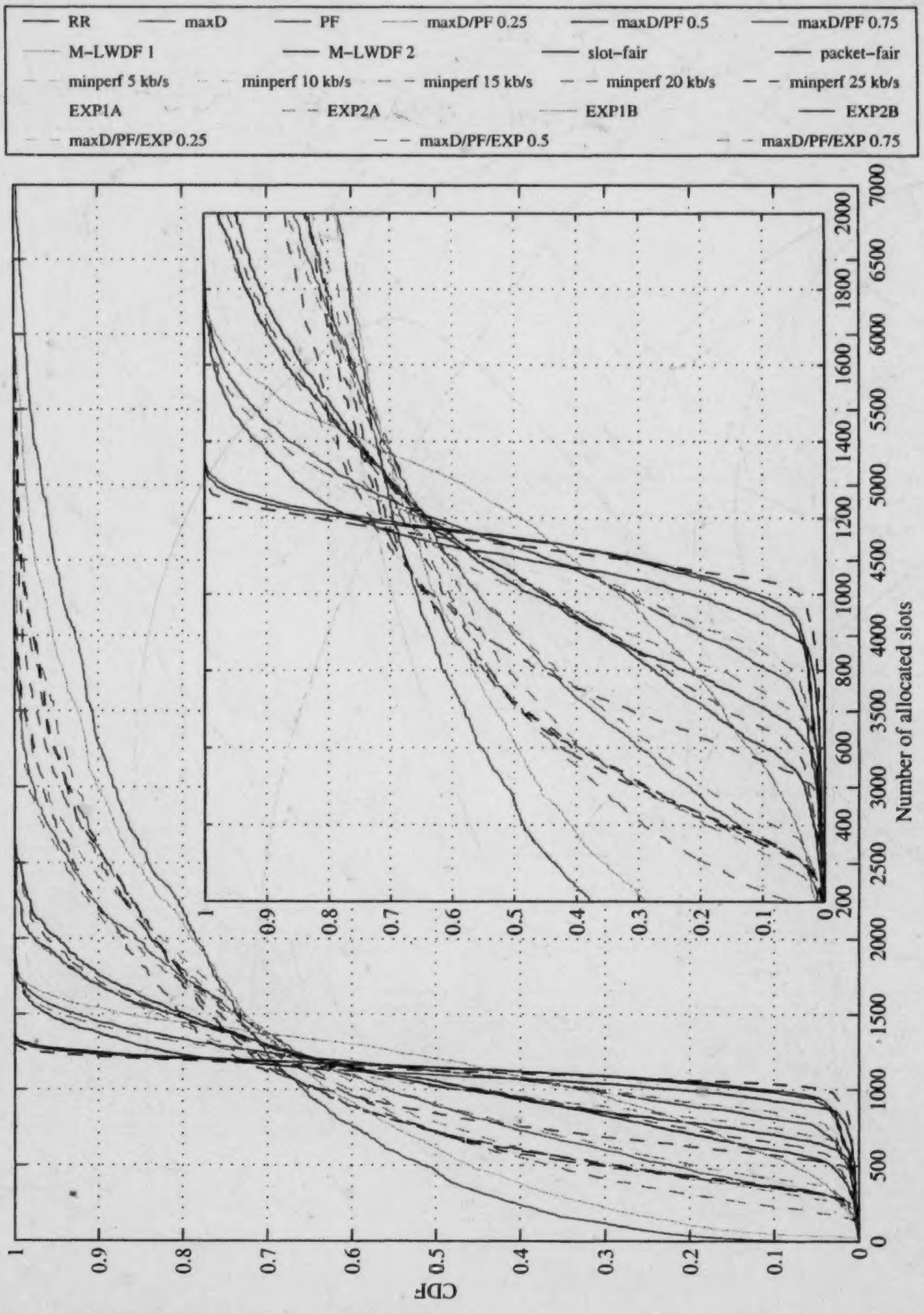


Figure I.10: Distributions of allocated slots per user per drop in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction

Appendix J: ITU Vehicular B Channel Results Comparing the "PF A" and "PF B" Average Rate Calculation Methods using Perfect Multislot Prediction

This appendix contains graphs comparing the simulation results using the "PF A" and "PF B" methods of calculating the average rate in the metrics of scheduling algorithms for the ITU vehicular B channel when perfect multislot prediction was used to determine the transmission format each user could support.

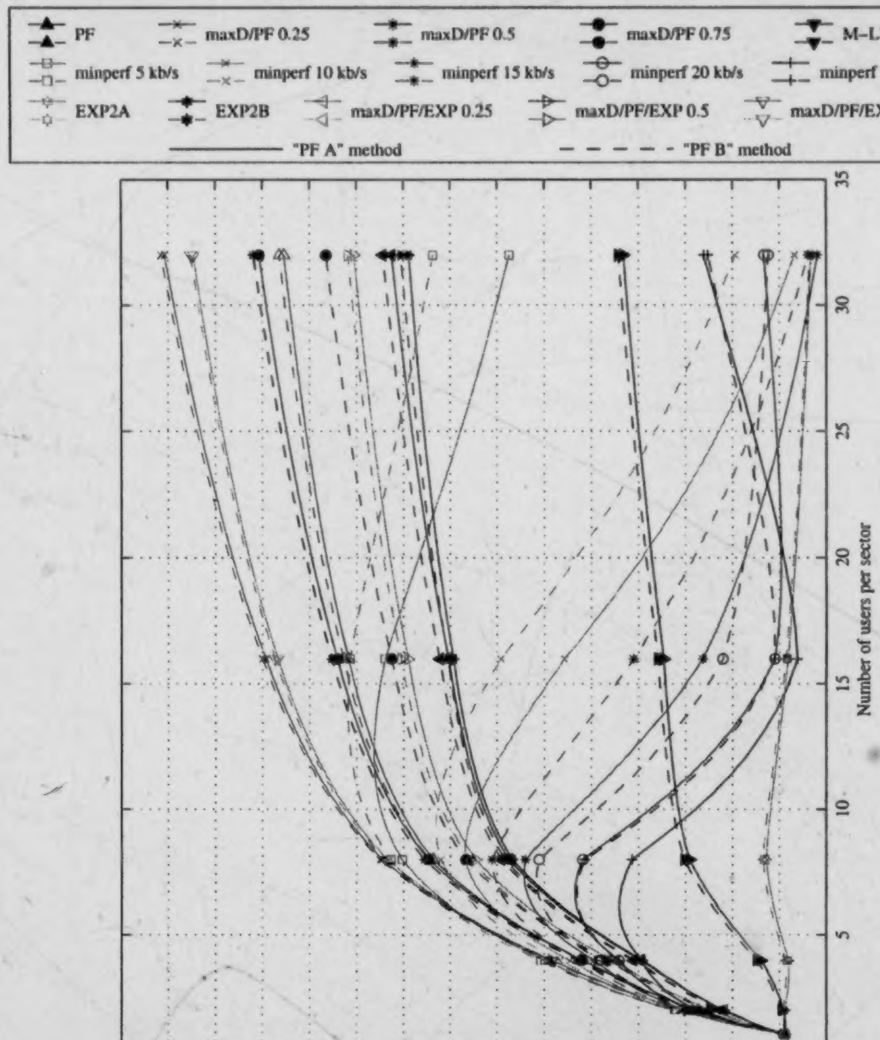


Figure J.1: Average sector throughput per drop vs. number of users per sector in the ITU vehicular B channel using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

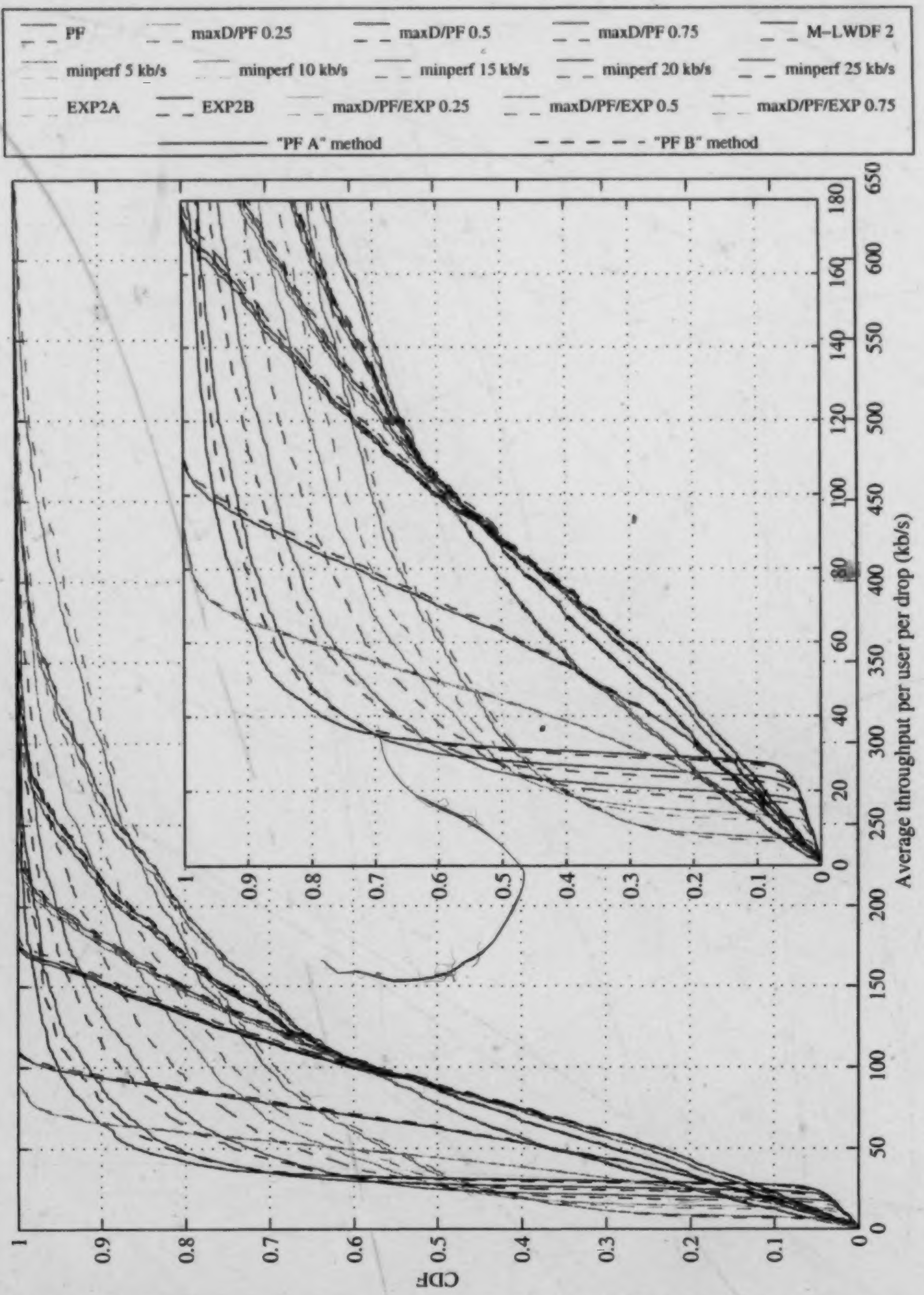


Figure J.2: Distributions of average throughput per user per drop in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

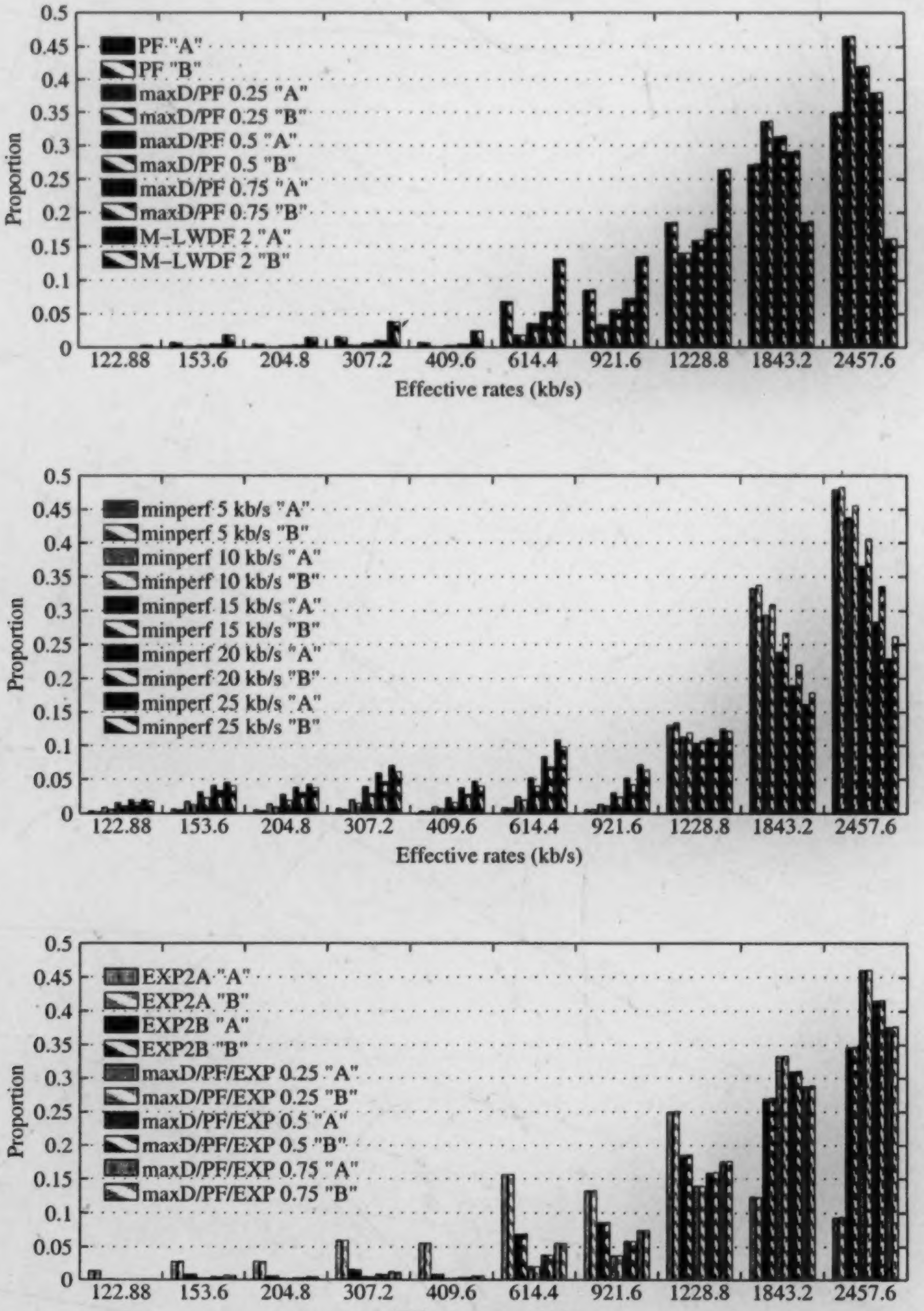


Figure J.3: Distribution of effective bit rates for transmitted packets in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

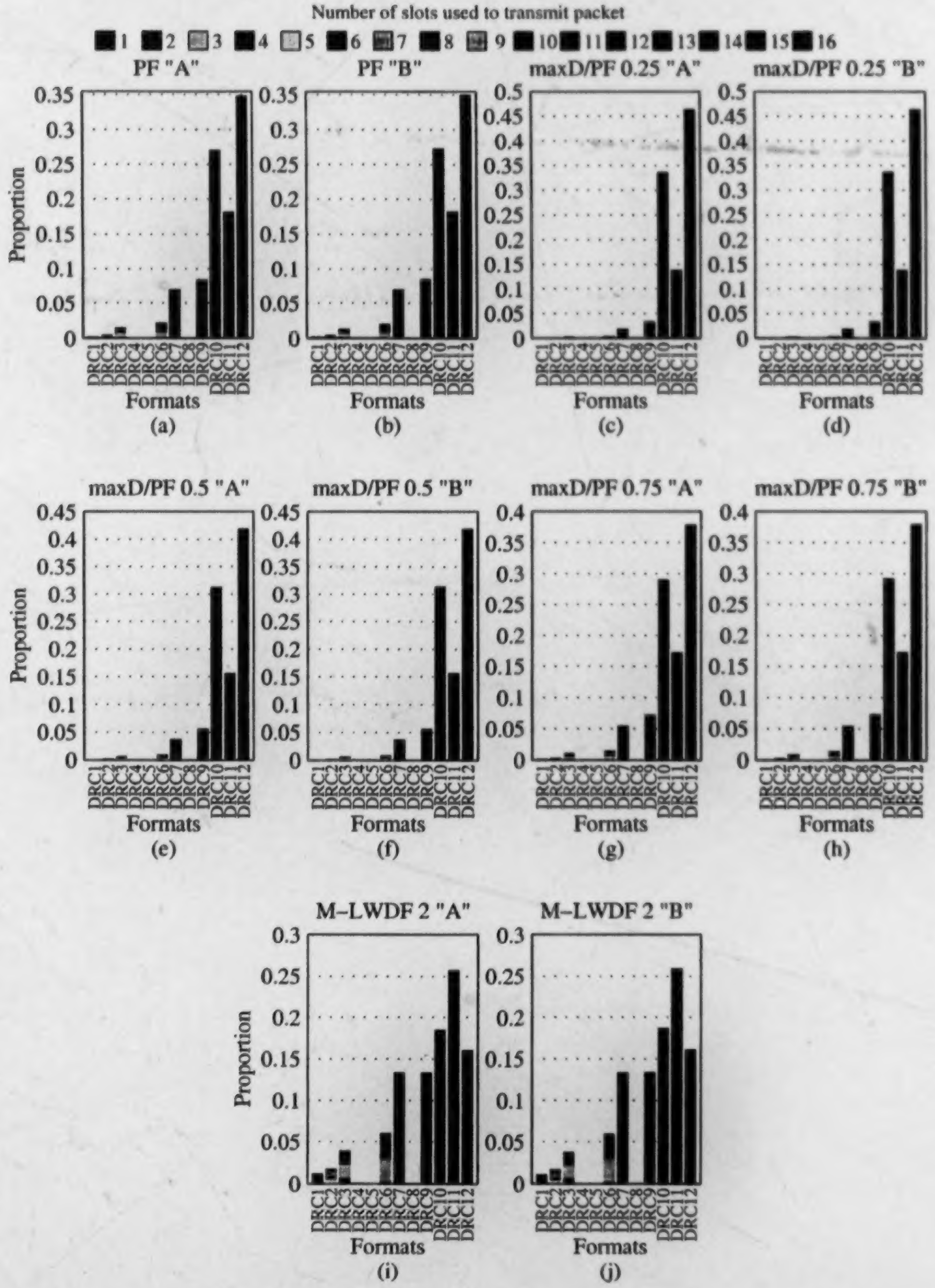


Figure J.4 (Part 1 of 3): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

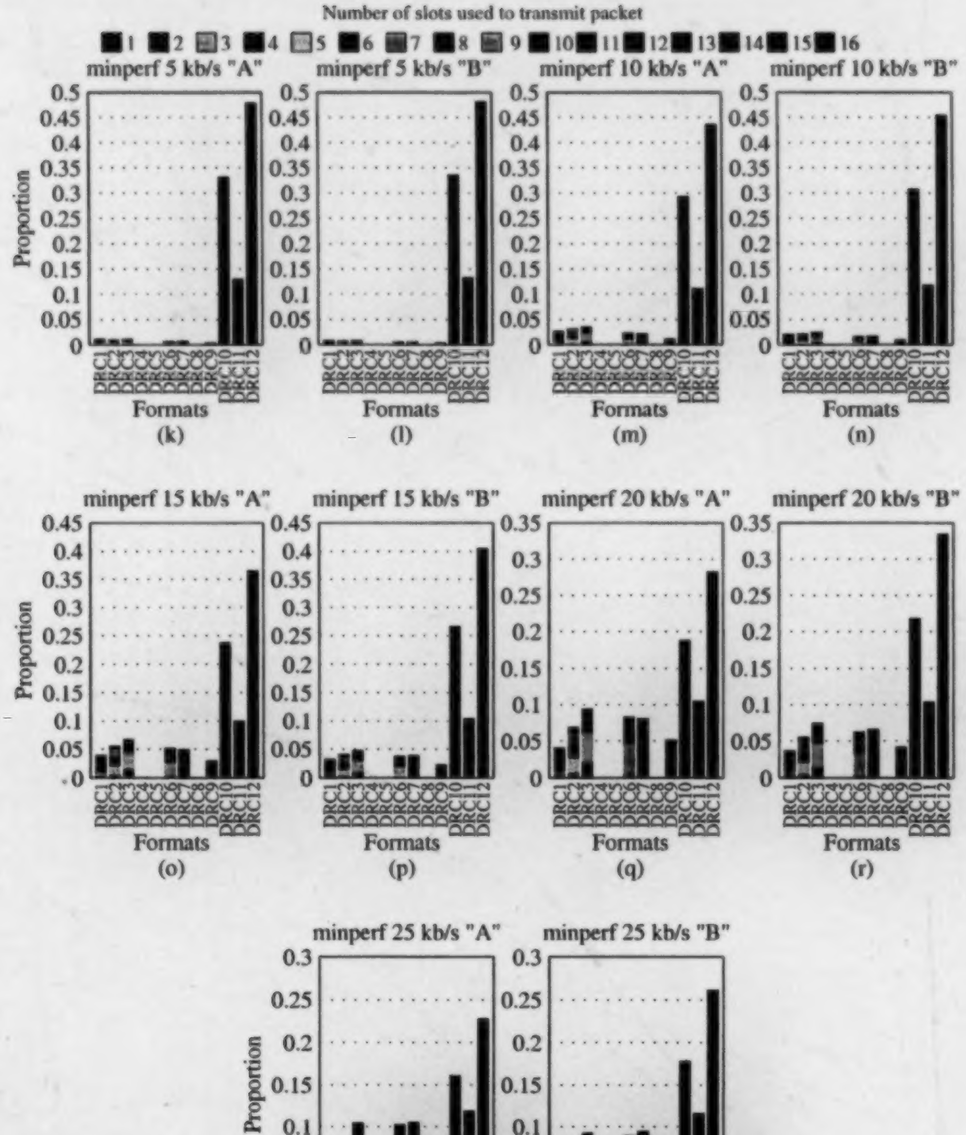


Figure J.4 (Part 2 of 3): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

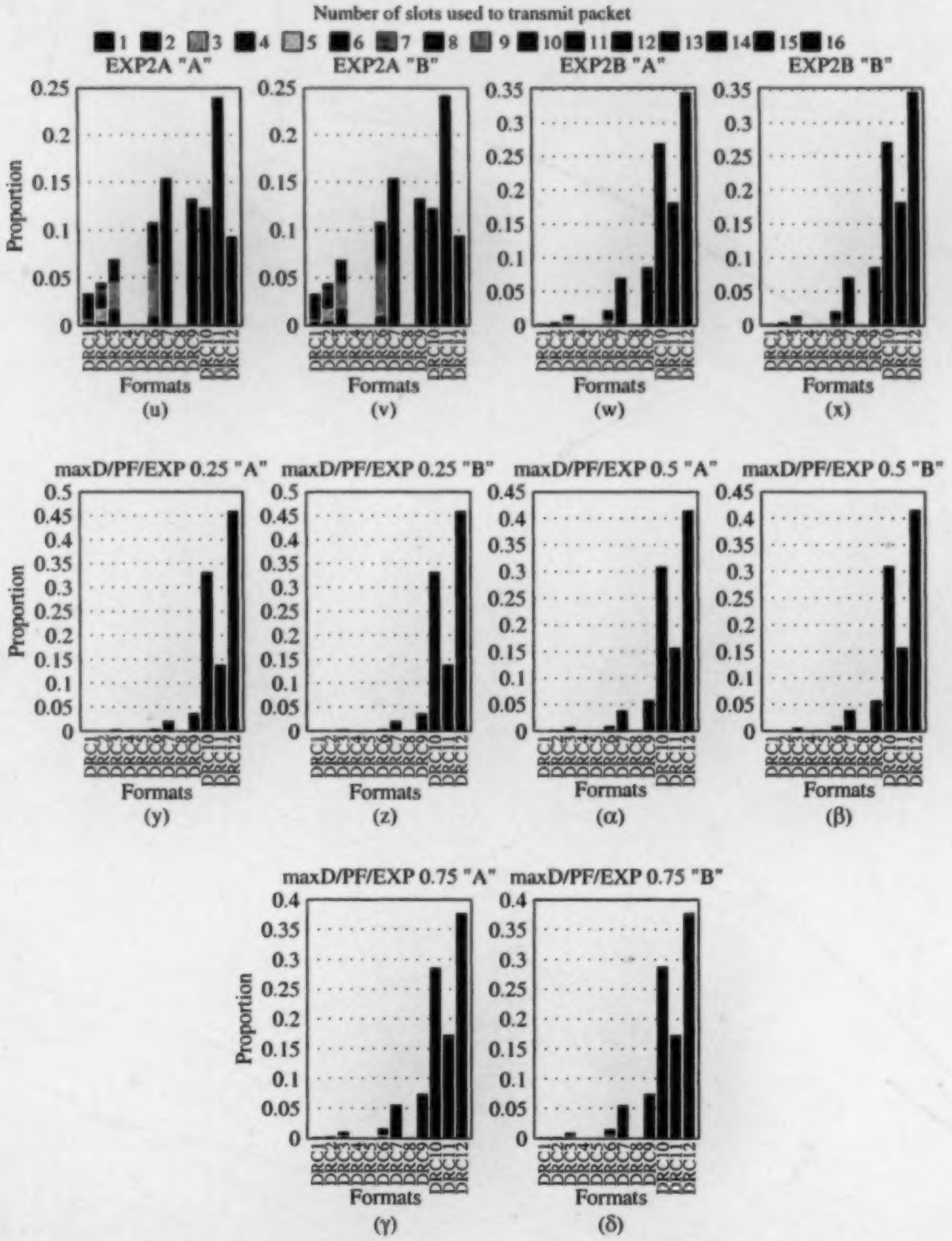


Figure J.4 (Part 3 of 3): Distribution of formats used for correctly received packets and number of slots used to receive the packets for scheduling algorithms in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

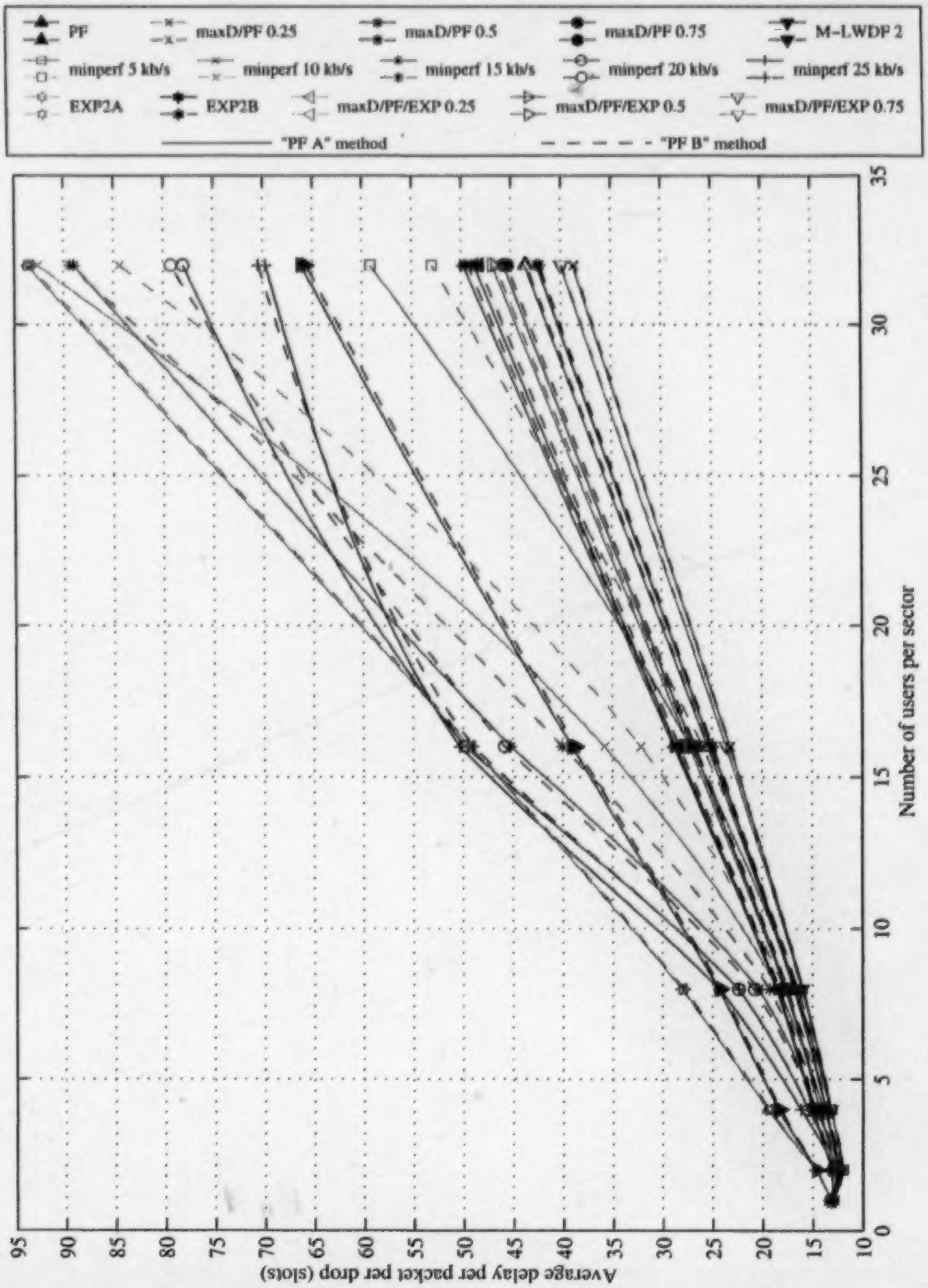


Figure J.5: Average delay per packet per drop versus number of users per sector in the ITU vehicular B channel using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

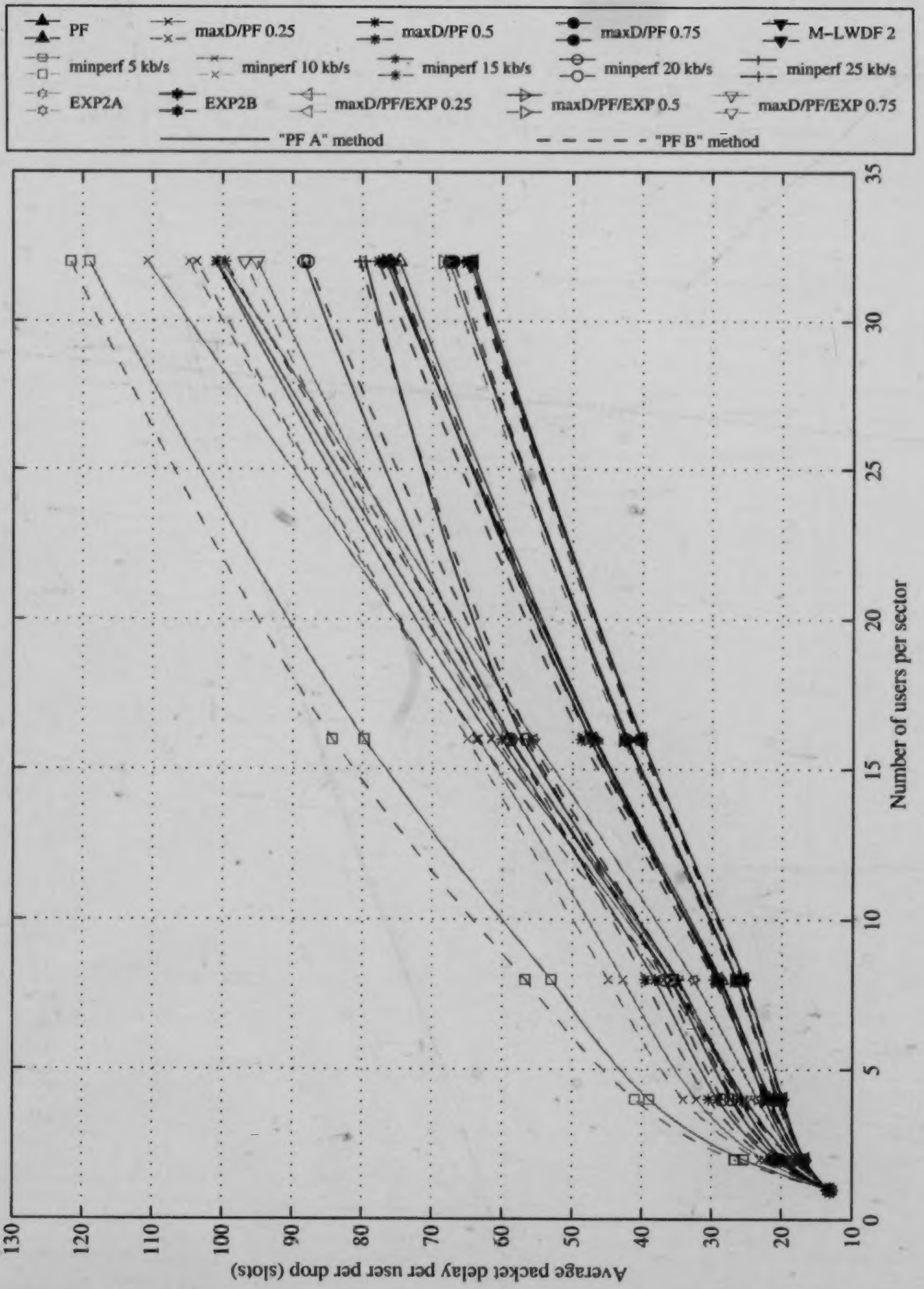


Figure J.6: Average packet delay per user per drop versus number of users per sector in the ITU vehicular B channel using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

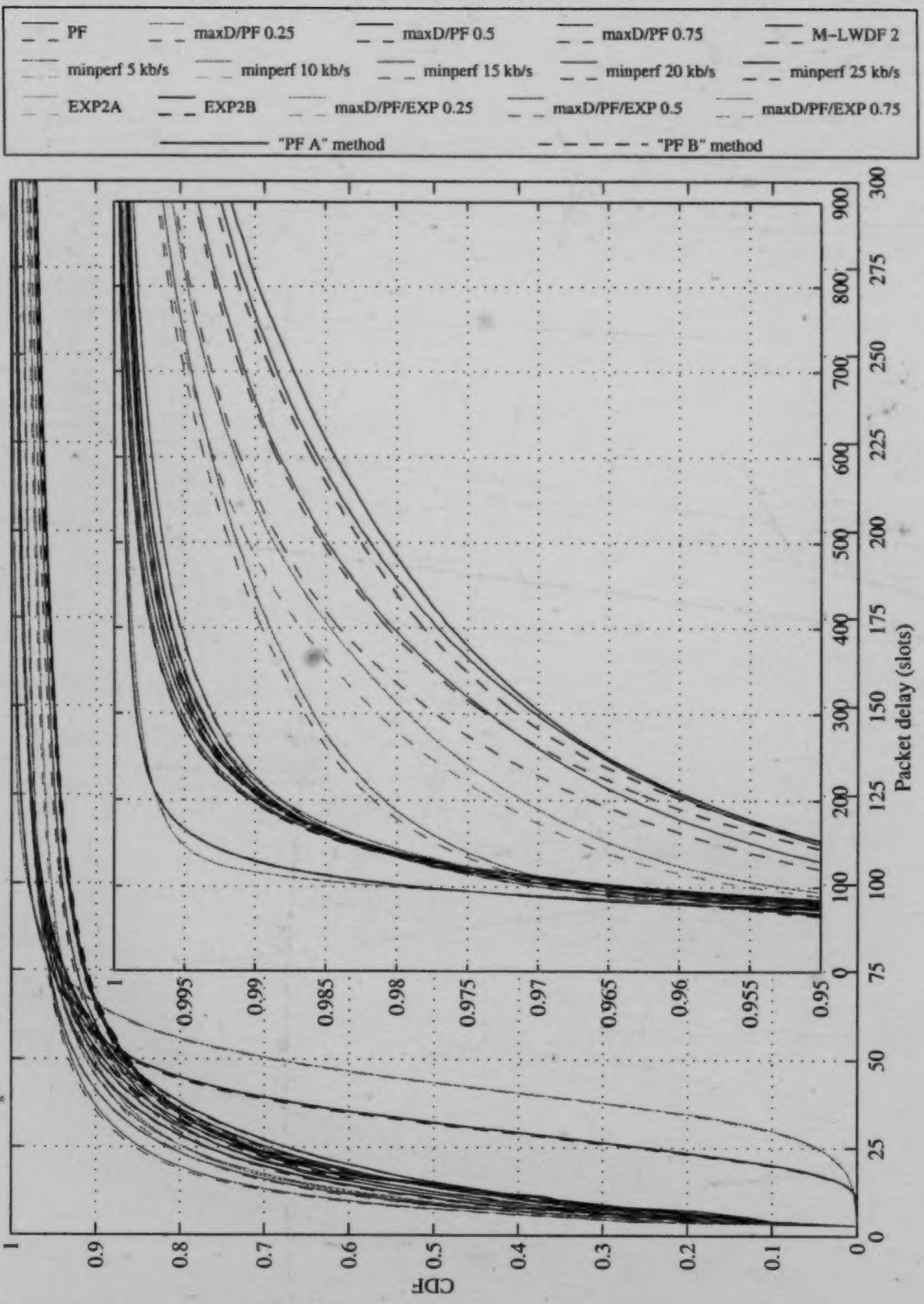


Figure J.7: Cumulative distribution functions of delays experienced by packets in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

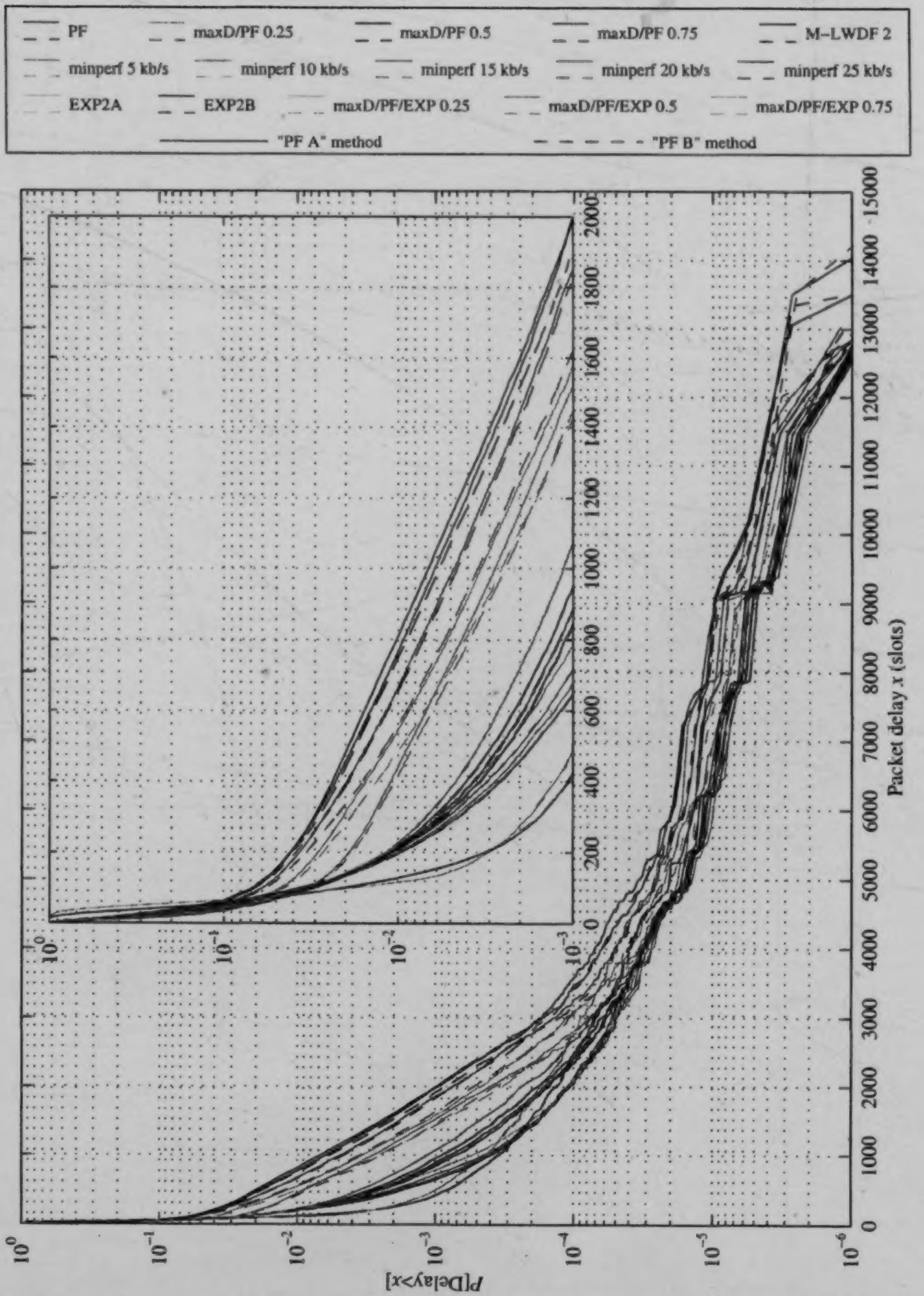


Figure J.8: Probability of delays experienced by packets being larger than a given value in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

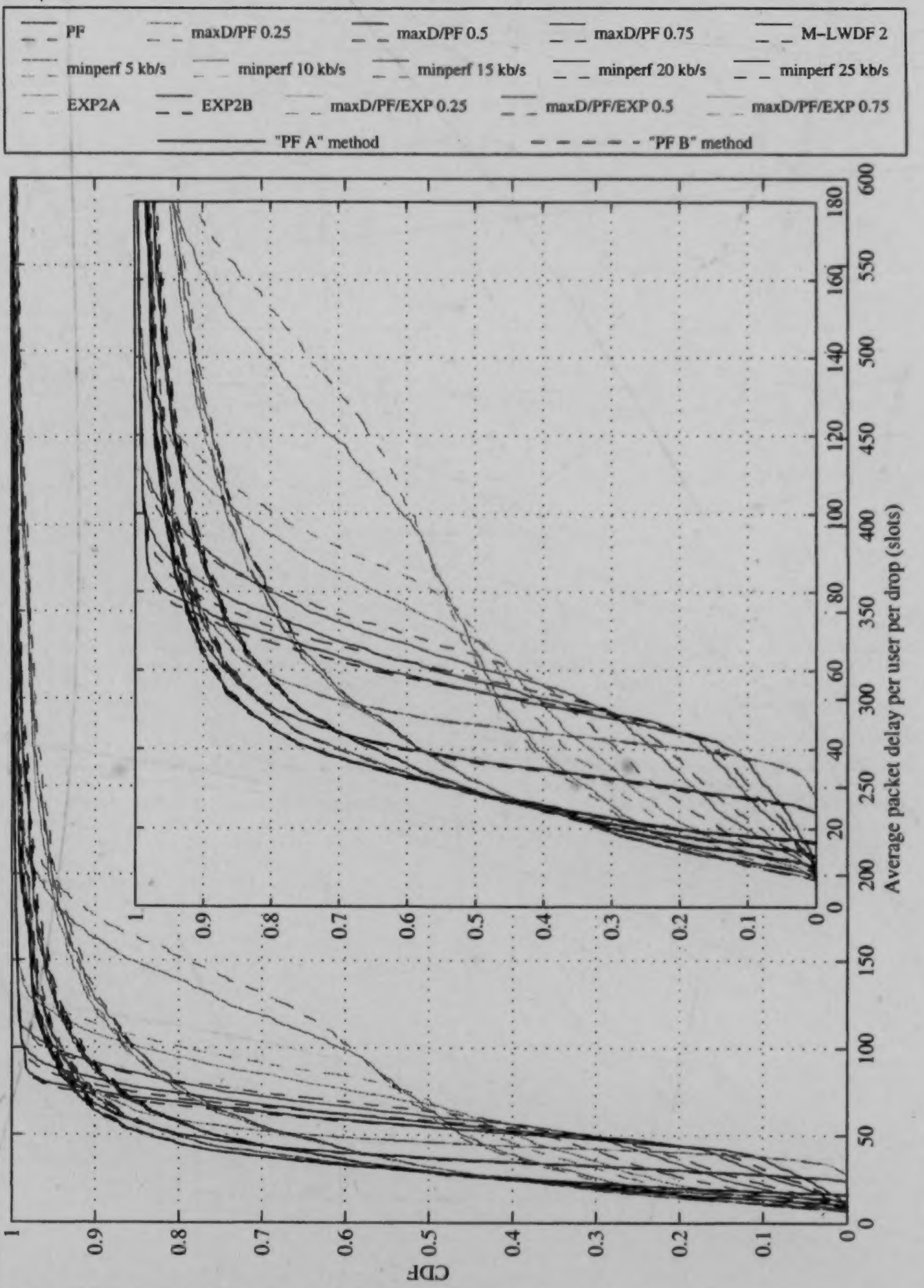


Figure J.9: Distributions of average packet delay per user for users who receive at least one packet in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

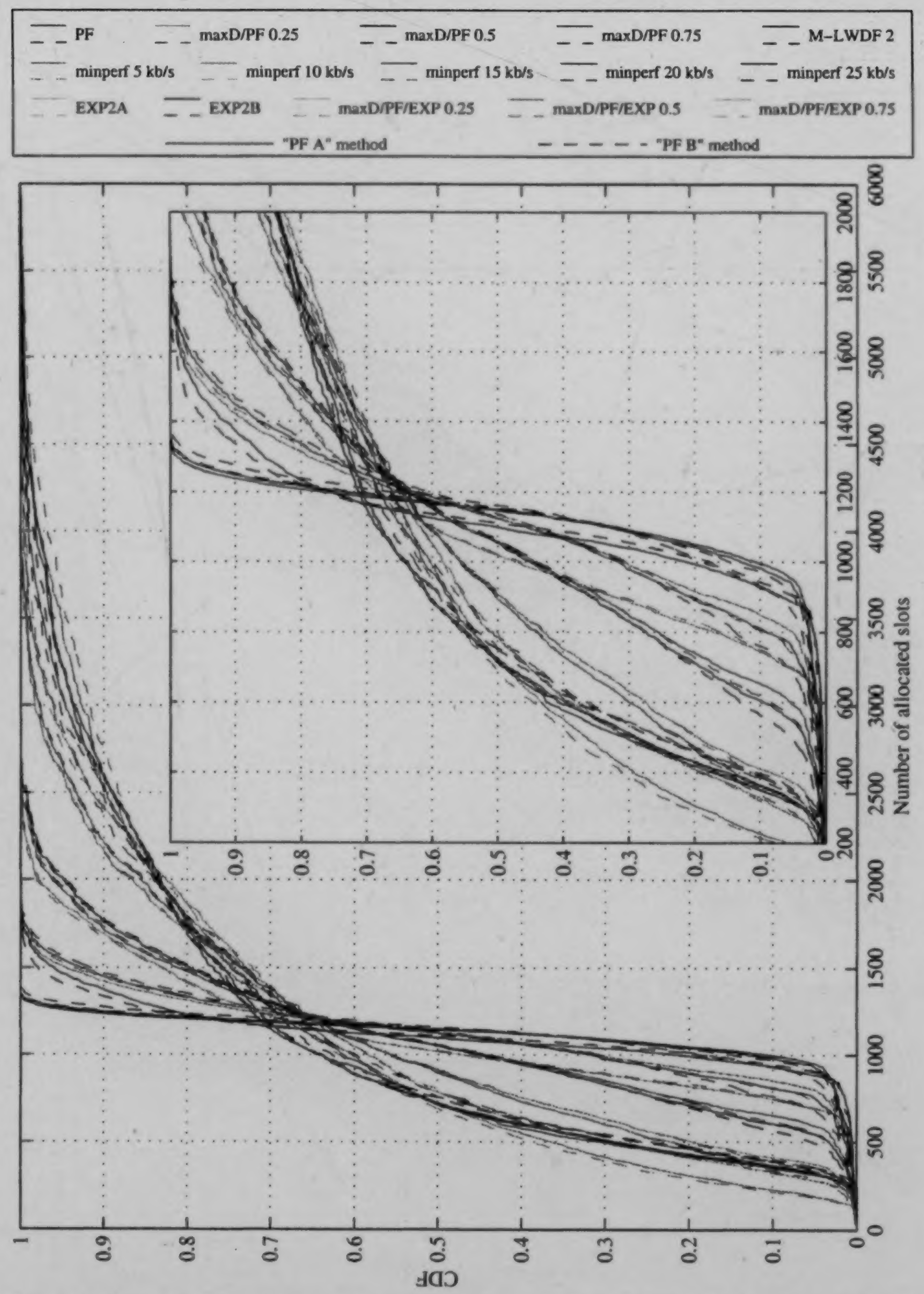


Figure J.10: Distributions of allocated slots per user per drop in the ITU vehicular B channel with 16 users per sector using perfect multislot prediction and comparing the "PF A" and "PF B" average rate calculation methods

Appendix K: Derivation of Probability Distribution of Users at a Given Distance from a Sector Vertex

Consider a hexagonal-shaped sector with users dropped uniformly over the area of the sector. Because of the uniform distribution, if a single user was dropped in the sector, and if one were to consider a small section of the sector, the probability of a user being located within that section would simply be the area of the section relative to the area of the entire sector.

Using this fact, the probability of a user being a given distance away from one of the vertices of the sector can be determined. Consider a section of a ring within the sector as shown in the figure below.

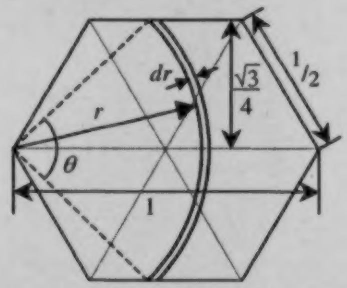


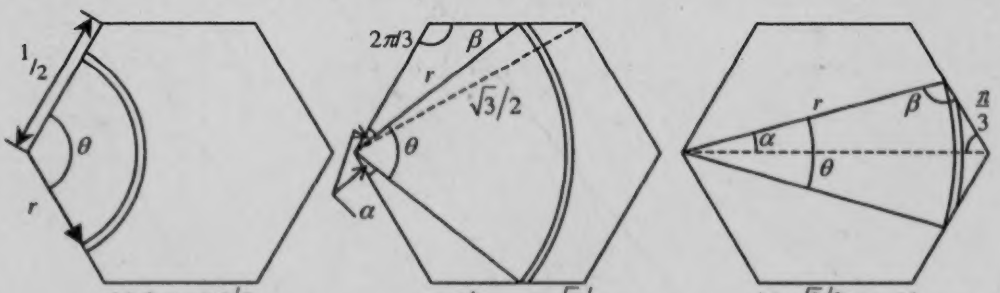
Figure K.1: An infinitesimal ring section within a hexagonal sector

The ring section has a radius *r*, has infinitesimal thickness *dr*, and encompasses an angle θ. The probability of a user being located at a distance *r* away from a corner of the hexagon is simply the area of the ring section relative to the total area of the hexagon. For the sake of simplicity, the radius of the sector (i.e. the distance between opposite corners of the sector) has been normalized to 1. The total area of the sector is then equal to 6 times the area of one of the small triangles that make up the hexagon as shown in the figure above.

$$A_{tot} = 6\left(\frac{1}{2}\cdot\frac{1}{2}\cdot\frac{\sqrt{3}}{4}\right) = \frac{3\sqrt{3}}{8} \qquad \text{(K.1)}$$

The area enclosed by the ring segment is simply $dA = \theta r dr$, where θ is in radians. The probability of a user being located at a radius *r* is therefore dA/A_{tot}.

Note that the angle θ is a function of the radius *r*. There are three possibilities to consider, as shown in the following figure.



(a) $0 \le r \le 1/2$ (b) $1/2 < r \le \sqrt{3}/2$ (c) $\sqrt{3}/2 < r \le 1$

Figure K.2: Three cases of the angle θ as a function of the radius r

In case (a) above, the angle θ is a constant of $2\pi/3$, so the ring segment encompasses an area of $dA_1 = (2\pi/3)rdr$.

In case (b), it is easiest to first find α as a function of r. By the sine law:

$$\frac{r}{\sin(2\pi/3)}=\frac{1/2}{\sin\beta}=\frac{1/2}{\sin\left(\pi-(2\pi/3+\alpha)\right)}=\frac{1/2}{\sin(2\pi/3+\alpha)} \qquad (\text{K.2})$$

The last term comes from the fact that $\sin(\pi - x) = \sin(x)$. We therefore have:

$$\sin\left(\frac{2\pi}{3}+\alpha\right)=\cos\left(\left(\frac{2\pi}{3}+\alpha\right)-\frac{\pi}{2}\right)=\cos\left(\frac{\pi}{6}+\alpha\right)=\frac{\sqrt{3}}{4r} \qquad (\text{K.3})$$

Therefore, we have $\alpha=-\pi/6+\arccos\left[\sqrt{3}/(4r)\right]$. In terms of θ, we then have:

$$\theta=\frac{2\pi}{3}-2\alpha=\frac{2\pi}{3}-2\left[\frac{-\pi}{6}+\arccos\left(\frac{\sqrt{3}}{4r}\right)\right]=\pi-2\arccos\left(\frac{\sqrt{3}}{4r}\right) \qquad (\text{K.4})$$

Thus, the area encompassed by the ring segment in case (b) is:

$$dA_2=\theta rdr=\left[\pi r-2r\arccos\left(\frac{\sqrt{3}}{4r}\right)\right]dr \qquad (\text{K.5})$$

By a similar nature, in case (c), we find:

$$\frac{r}{\sin(\pi/3)}=\frac{1}{\sin\beta}=\frac{1}{\sin\left(\pi-(\pi/3+\alpha)\right)}=\frac{1}{\sin(\pi/3+\alpha)}\Rightarrow\sin\left(\frac{\pi}{3}+\alpha\right)=\frac{\sqrt{3}}{2r} \qquad (\text{K.6})$$

This gives us $\alpha=-\pi/3+\arcsin\left[\sqrt{3}/(2r)\right]$, and hence:

$$dA_3=\theta rdr=(2\alpha)rdr=\left[\frac{-2\pi}{3}r+2r\arcsin\left(\frac{\sqrt{3}}{2r}\right)\right]dr \qquad (\text{K.7})$$

To find the total area inside a given radius, the above areas inside the ring segments are integrated. In case (a), this gives:

$$0 \le r \le 1/2: \quad A_1 = \int_0^r \frac{2\pi}{3} r\,dr = \frac{\pi}{3} r^2 \Big|_0^r = \frac{\pi}{3} r^2 \qquad (\text{K.8})$$

At $r = ½$, we get $A_1 = \pi/3 \times (½)^2 = \pi/12$.

In case (b), we get:

$$\frac{1}{2} \le r \le \frac{\sqrt{3}}{2}: \quad A_2 = \int_{1/2}^r \left[\pi r - 2r \arccos\left(\frac{\sqrt{3}}{4r}\right) \right] dr \qquad (\text{K.9})$$

Using MATLAB, it was found that the integral of the above equation is:

$$\begin{aligned} A_2 &= \left[\frac{1}{2}\pi r^2 - r^2 \arccos\left(\frac{\sqrt{3}}{4r}\right) + \frac{\sqrt{3}r}{4}\sqrt{1 - \frac{3}{16r^2}} \right]_{1/2}^{r} \\ &= \frac{1}{2}\pi r^2 - r^2 \arccos\left(\frac{\sqrt{3}}{4r}\right) + \frac{\sqrt{3}r}{4}\sqrt{1 - \frac{3}{16r^2}} \\ &\quad - \frac{1}{2}\pi\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 \arccos\left(\frac{\sqrt{3}}{4(1/2)}\right) - \frac{\sqrt{3}(1/2)}{4}\sqrt{1 - \frac{3}{16(1/2)^2}} \\ &= \frac{1}{2}\pi r^2 - r^2 \arccos\left(\frac{\sqrt{3}}{4r}\right) + \frac{\sqrt{3}r}{4}\sqrt{1 - \frac{3}{16r^2}} - \frac{\pi}{12} - \frac{\sqrt{3}}{16} \end{aligned} \qquad (\text{K.10})$$

At $r = \sqrt{3}/2$, this reduces to $A_2 = \pi/24 + \sqrt{3}/8$.

Case (c) gives:

$$\frac{\sqrt{3}}{2} < r \le 1: \quad A_3 = \int_{\sqrt{3}/2}^r \left[\frac{-2\pi}{3} r + 2r \arcsin\left(\frac{\sqrt{3}}{2r}\right) \right] dr \qquad (\text{K.11})$$

Once again using MATLAB, we get:

$$\begin{aligned} A_3 &= \left[\frac{-\pi}{3}r^2 + r^2\arcsin\left(\frac{\sqrt{3}}{2r}\right) + \frac{\sqrt{3}r}{2}\sqrt{1-\frac{3}{4r^2}}\,\right]_{\sqrt{3}/2}^{r} \\ &= \frac{-\pi}{3}r^2 + r^2\arcsin\left(\frac{\sqrt{3}}{2r}\right) + \frac{\sqrt{3}r}{2}\sqrt{1-\frac{3}{4r^2}} \\ &\quad + \frac{\pi}{3}\left(\frac{\sqrt{3}}{2}\right)^2 - \left(\frac{\sqrt{3}}{2}\right)^2 \arcsin\left(\frac{\sqrt{3}}{2}\cdot\frac{2}{\sqrt{3}}\right) - \frac{\sqrt{3}}{2}\cdot\left(\frac{\sqrt{3}}{2}\right)\sqrt{1-\frac{3}{4}\cdot\left(\frac{2}{\sqrt{3}}\right)^2} \\ &= \frac{-\pi}{3}r^2 + r^2\arcsin\left(\frac{\sqrt{3}}{2r}\right) + \frac{\sqrt{3}r}{2}\sqrt{1-\frac{3}{4r^2}} - \frac{\pi}{8} \end{aligned} \tag{K.12}$$

Substituting $r = 1$ yields $A_3 = \sqrt{3}/4 - \pi/8$.

As a check, the total area would be given by A_1+A_2+A_3, which is:

$$A_1 + A_2 + A_3 = \left(\frac{\pi}{12}\right) + \left(\frac{\pi}{24} + \frac{\sqrt{3}}{8}\right) + \left(\frac{\sqrt{3}}{4} - \frac{\pi}{8}\right) = \frac{3\sqrt{3}}{8} \tag{K.13}$$

This is the same value as calculated before.

The area within a given radius as a function of r would be given by:

$$A(r) = \begin{cases} \int_0^r dA_1, & 0 \le r \le 1/2 \\ \int_0^{1/2} dA_1 + \int_{1/2}^r dA_2, & 1/2 < r \le \sqrt{3}/2 \\ \int_0^{1/2} dA_1 + \int_{1/2}^{\sqrt{3}/2} dA_2 + \int_{\sqrt{3}/2}^r dA_3, & \sqrt{3}/2 < r \le 1 \end{cases} \tag{K.14}$$

The probability of a user being located within a given radius is the cumulative density function (CDF) of the user's radius, and can be found by calculating $A(r)/A_{tot}$:

$$CDF(r) = \begin{cases} \frac{8\pi}{9\sqrt{3}}r^2, & 0 \le r \le \frac{1}{2} \\ \frac{4\pi}{3\sqrt{3}}r^2 - \frac{8}{3\sqrt{3}}r^2\arccos\left(\frac{\sqrt{3}}{4r}\right) + \frac{2}{3}r\sqrt{1-\frac{3}{16r^2}} - \frac{1}{6}, & \frac{1}{2} < r \le \frac{\sqrt{3}}{2} \\ \frac{-8\pi}{9\sqrt{3}}r^2 + \frac{8}{3\sqrt{3}}r^2\arcsin\left(\frac{\sqrt{3}}{2r}\right) + \frac{4}{3}r\sqrt{1-\frac{3}{4r^2}} + \frac{1}{3}, & \frac{\sqrt{3}}{2} < r \le 1 \end{cases} \tag{K.15}$$

The probability density function (PDF) is the derivative of the CDF with respect to r. It can be found quite simply by dividing the original terms for dA_1, dA_2, and dA_3 (with the "dr" removed) by A_{tot}:

$$PDF(r)=\begin{cases}\dfrac{16\pi}{9\sqrt{3}}r, & 0\le r\le\dfrac{1}{2}\\ \dfrac{8\pi}{3\sqrt{3}}r-\dfrac{16}{3\sqrt{3}}r\arccos\left(\dfrac{\sqrt{3}}{4r}\right), & \dfrac{1}{2}<r\le\dfrac{\sqrt{3}}{2}\\ \dfrac{-16\pi}{9\sqrt{3}}r+\dfrac{16}{3\sqrt{3}}r\arcsin\left(\dfrac{\sqrt{3}}{2r}\right), & \dfrac{\sqrt{3}}{2}<r\le 1\end{cases}\tag{K.16}$$

Therefore, equations expressing the probability of a user being located at a distance *r* or within a distance *r* from a vertex of the sector have been derived. The probability of a user being located outside a distance *r* from a given vertex would be 1 minus the *CDF*(*r*). The following graph shows the PDF and CDF of a user being located a distance *r* away from a vertex of the sector.

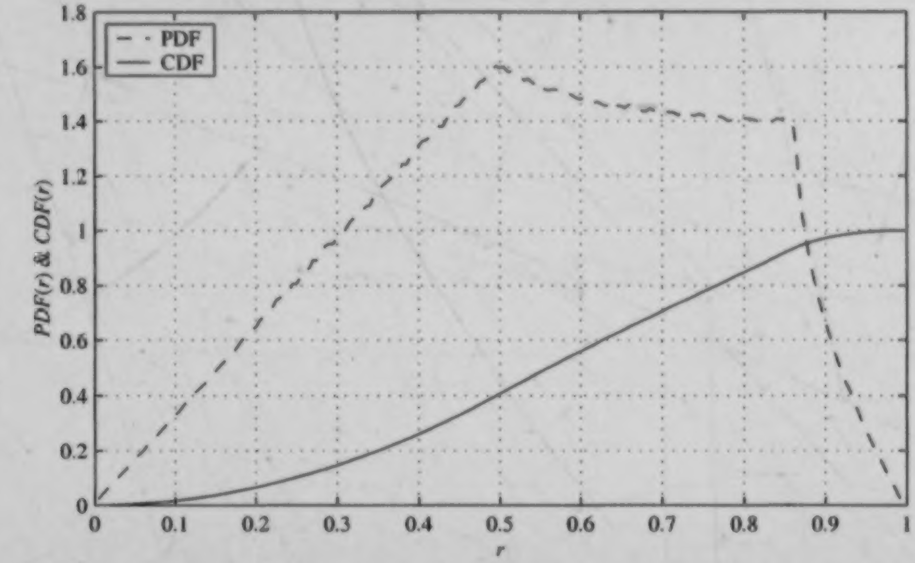


Figure K.3: PDF and CDF distributions of a user being located a distance *r* from the vertex of a hexagonally-shaped sector, given that the user is located randomly and uniformly over the area of the sector

What has been examined thus far is the case of one user being located in the sector. It shall now be examined what happens when more than one user is located in the sector.

First, consider the case of two users per sector. The probability of both users being located within a given radius *r* is simply $[CDF(r)]^2$. Similarly, the probability of both users being outside that radius is $[1 - CDF(r)]^2$. However, the probability of one user being inside the radius and the other outside is in fact $2\times[CDF(r)][1 - CDF(r)]$. The factor of 2 stems from the fact that there are two possible ways of this latter scenario occurring: either user 1 can be inside the radius and user 2 outside, or user 2 could be inside and user 1 outside.

More generally, if there are N users in the sector, the probability of x users (with $0 \leq x \leq N$) being inside the given radius, and $N-x$ users being outside the radius is:

$$P(x \text{ users inside radius } r, N \text{ users outside radius } r) = \left[CDF(r)\right]^{x}\left[1-CDF(r)\right]^{N-x}\binom{N}{x},$$

$$\text{where } \binom{N}{x} \text{ is the "}N\text{ choose }x\text{" operator, i.e. } \binom{N}{x} = \binom{N}{N-x} = \frac{N!}{x!(N-x)!} \qquad (K.17)$$

From this, the probability of at least n out of N users being outside a given radius r can be expressed as:

$$P(\text{at least } n \text{ outside of } r) = \sum_{y=n}^{N}\binom{N}{y}\left[CDF(r)\right]^{N-y}\left[1-CDF(r)\right]^{y} \qquad (K.18)$$

The following graph shows the probability of n out of N users being outside of a given radius r for several combinations of n and N.

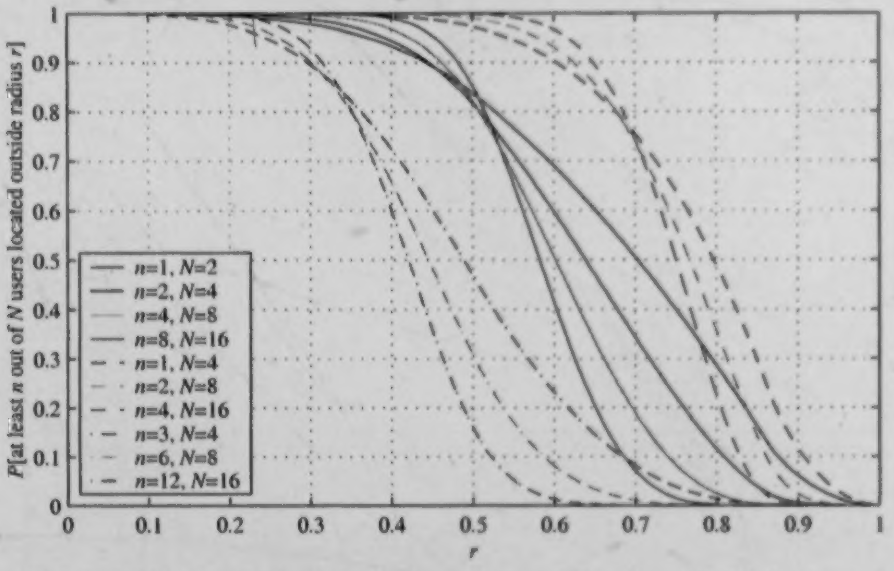


Figure K.4: Probability distributions of at least n users out of N total per sector being located at a distance larger than a radius r from the vertex of the sector

Now, consider what happens at the upper percentiles of the distributions. A closer look at that area is provided in the graph below.

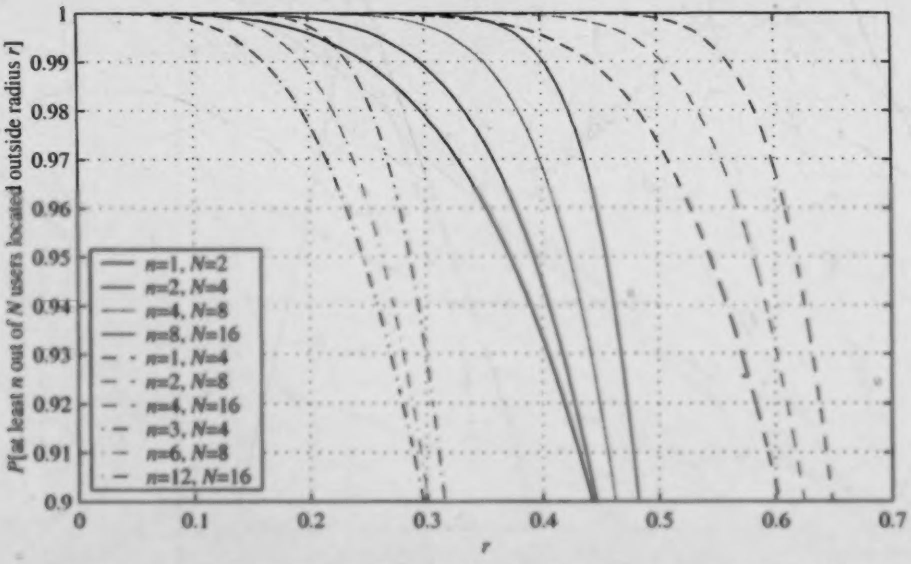


Figure K.5: Upper percentiles for probability distributions of at least *n* users out of *N* total per sector being located at a distance larger than a radius *r* from the vertex of the sector

For 2 users per sector, 95% of the time (i.e. 19 times out of 20), at least half of the users will be located outside a radius of 0.324. For 4 users per sector, 95% of the time at least half of the users will be located outside a radius of 0.393. For 8 and 16 users per sector, the respective radii are 0.424 and 0.455. If we switch to the 99th percentile (i.e. 99 times out of 100), for 2, 4, 8, and 16 users per sector, at least half the users will be located outside radii of 0.249, 0.296, 0.351, and 0.402. The following table compares the resulting radii if proportions of at least 25% of the users and at least 75% of the users.

TABLE LXXXVII:
P PERCENT OF THE TIME, AT LEAST *Q* PERCENT OF THE USERS OUT OF *N* USERS IN TOTAL WILL BE LOCATED OUTSIDE A RADIUS OF *r* AWAY FROM THE VERTEX OF THE SECTOR

Percentile *P*	95%									
Proportion *Q*	25%			50%				75%		
Number of Users *N*	4	8	16	2	4	8	16	4	8	16
Radius *r*	0.544	0.582	0.618	0.324	0.393	0.424	0.455	0.246	0.263	0.286
Percentile *P*	99%									
Proportion *Q*	25%			50%				75%		
Number of Users *N*	4	8	16	2	4	8	16	4	8	16
Radius *r*	0.443	0.504	0.560	0.249	0.296	0.351	0.402	0.161	0.194	0.233

Notice that for a given percentile and proportion, the radius outside which one can most of the time expect to find a certain proportion of the users increases as the number of users per sector increases. This therefore demonstrates that the larger the number of users with a sector, any given proportion of users will be located further away from the transmitter than if there were fewer users per sector. Alternately, for a given radius, most of the time, a larger proportion of the users can be expected to be outside that radius for larger numbers of users per sector. Hence, this explains why an system using an algorithm such as round robin that does not take channel conditions into account in its scheduling decision but rather chooses users in cyclic order or at random will experience a decrease in throughput as the number of users per sector increases if the users are distributed uniformly over the area of the sector.